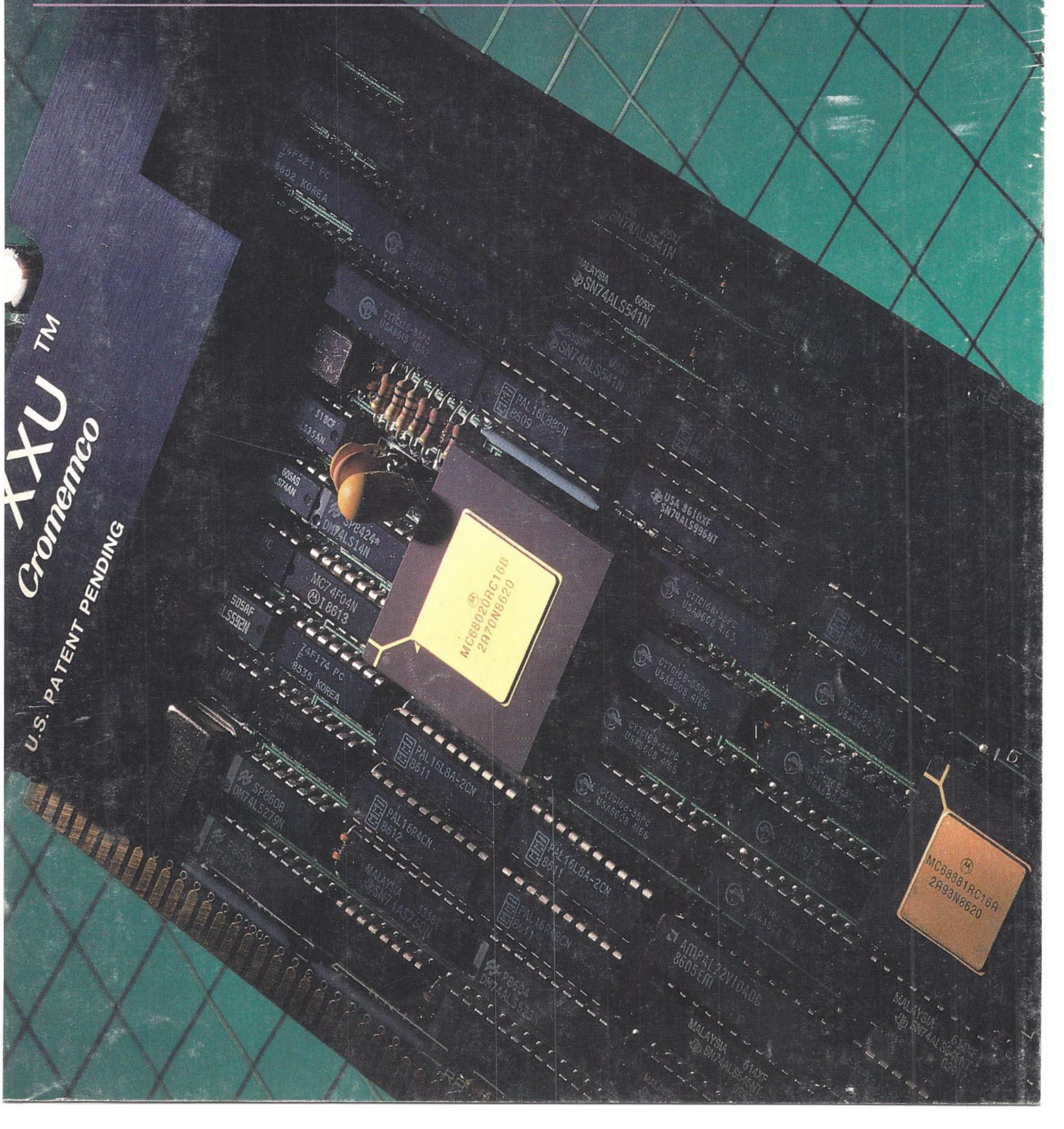
$4.90

VOL. 1, NO. 4 • FALL 1986

S-100 journal


XXU TM
Cromemco
U.S. PATENT PENDING
MC68020RC16B
2A70N8620
MC68881RC16A
2A93N8620

NOVELL COMPATIBLE

ONE-STOP ARCnet™ LAN SOLUTIONS

Now you can get all the tools you need to build industry standard **TOKEN-PASSING** networks from one source.

InterContinental Micro.

Our networking packages let you configure any combination of nodes (PC s and compatibles, ATs, XTs, Jrs., Z-100 s) to a common Fileserver in just a few minutes with our menu-driven installation program.

Active and passive hubs allow you to integrate up to 255 users on a single network with *increased* efficiency as the network grows.

In addition, our stand-alone ARCnet™ Workstation/Fileserver and complete line of S-100 Bus products (single board computers, slaves and controllers) give you and your customers even more options.

ARCnet is a trademark of Datapoint Corporation.

We're also compatible with other industry-leading ARCnet PC™ and S-100 products, and our PC networking boards include an optional 256K of system RAM.

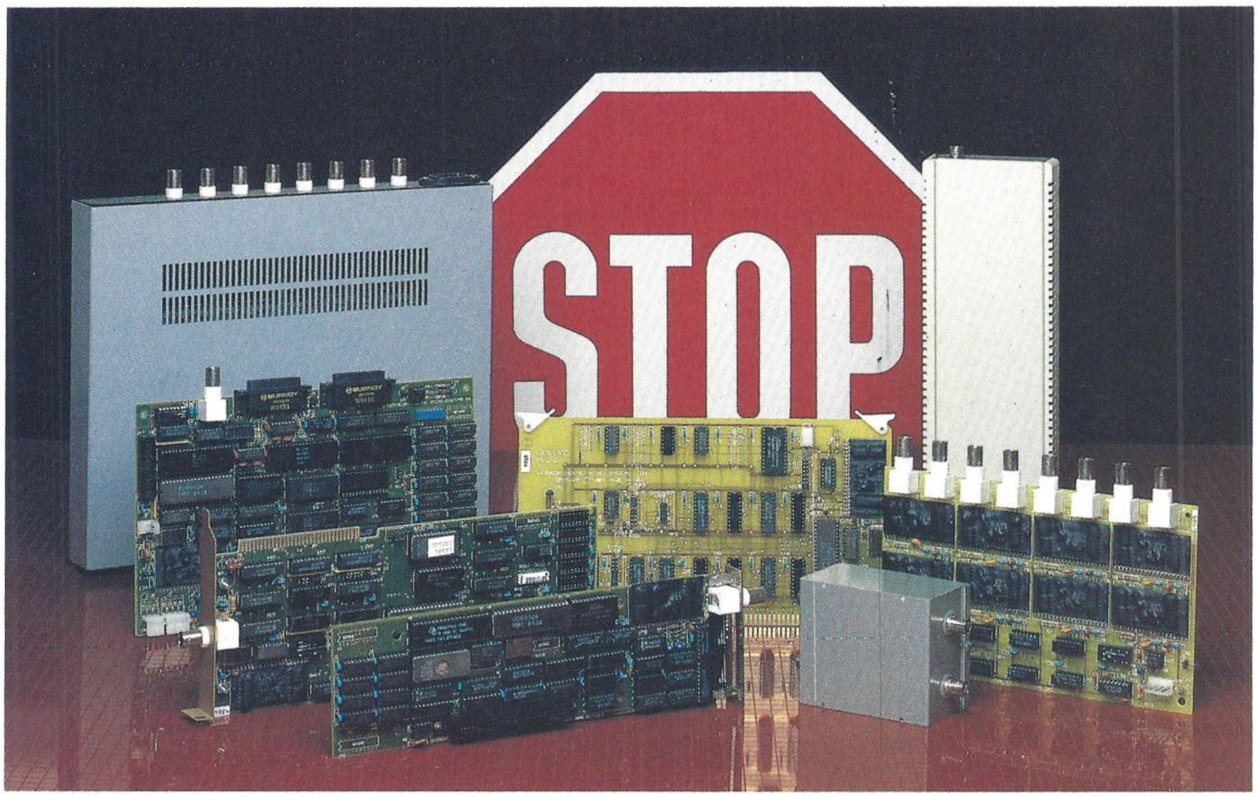
All with InterContinental Micro's reputation for outstanding reliability, exceptional service, and highly competitive pricing.

Call InterContinental Micro today, or circle the bingo number below for ARCnet™ LAN solutions.

InterContinental Micro

4015 Leaverton Court, Anaheim, CA
92807; Phone: (714) 630-0964
Telex: 821375 SUPPORT UD: Easylink: 62562040.

Call for GSA pricing.



S-100 journal

VOL. 1 NO. 4 FALL 1986

Editor and Designer
Jay Vilhena

Associate Editor
Robert Petersen

Technical Editor
Don Pannell

Circulation
Linda Vilhena

Advertising
Jay Vilhena

Software Consultant
Steve Renshaw

Contributing Artists
Joslyn L. Johanna
G. N. Pereira

S-100 Journal is published quarterly by Octoplus Corporation, 1275 N. University Ave., Unit 7, Provo, UT 84604. Telephone (801) 373-0696.

Copyright©1986 by Octoplus Corporation. All rights reserved.

Subscriptions: U.S.A.: $14.00 for 1 year, $25.00 for 2 years. Canada and Mexico: add $3.00 per year. Other countries: add $6.00 per year. All foreign payments must be in U.S. currency. Send orders and payments to S-100 Journal, P.O. Box 1914, Orem, UT 84057.
Advertising: Contact Jay Vilhena at one of above addresses or call (801) 373-0696.
Postmaster: Send address changes to S-100 Journal, P.O. Box 1914, Orem, UT 84057.

outline

23 LOCAL AREA NETWORKS UNDER TurboDOS

38 BUILD A 4-PORT SERIAL S-100 BOARD

62 32 BITS ON THE S-100 BUS

69 THE CROMEMCO XXU A 32-BIT S-100 BOARD

4 EDITORIAL
7 EDITOR INTERFACE
8 696 BUS
13 MULTIUSER OS
59 BITS
72 NEWS AND NEW PRODUCTS
75 S-100 SOFTWARE
76 S-100 DIRECTORY

OUR COVER

The new XXU, a 68020/68881 CPU card from Cromemco, delivers true 32-bit processing for the S-100 bus. See page 69.

Cover photograph by Mark Gottlieb. Photograph production courtesy of Cromemco, Inc.

What you should know about the S-100 market & Advanced Digital... BEFORE you buy;

"With the new wave of outstanding S-100 products hitting the market, S-100 manufacturers continue to assert their dedication to superior quality and innovative design. For many years ahead, they will equip thousands of small businesses and professionals with the most versatile and upgradable microsystems in existence."

—Jay Vilhena, Editor
S-100 Journal

"At Advanced Digital Corporation our business philosophy can be stated in one word - "Pride." Pride in the quality and performance of the products we design and manufacture...Pride in the dedication, experience and performance of our employees...and Pride in our achievements and contributions to the S-100 marketplace."

—Hossein Asadi, President
Advanced Digital Corporation

1. Attractive Features. Attractive Prices.

Advanced Digital Corporation offers the most complete line of S-100 products with the latest in the state-of-the-art microcomputer technology. All of ADC's S-100 products are powerful high speed 8-bit and 16-bit single board computers. Your S-100 system can grow from ordinary to awesome with ADC's superb family of S-100 boards quickly, easily and cost-effectively.

2. Financial Stability

Founded in 1980, Advanced Digital Corporation is one of the most financially sound, privately held computer manufacturers. The company has consistently shown a profit in every quarter since 1980. It is the intention of ADC to continue as the industry leader and remain in the forefront of technology assuring you of the finest product and service for years to come.

3. Quality Assurance

Rigorous testing of each and every product by ADC's highly skilled technicians assures customers of trouble-free products. ADC prides itself with the strictest quality control in the industry. No company has higher quality control standards, or a better success rate than ADC's. Every product is backed by a one year warranty.

4. Customer Service; 24 Hour Repair.

Highly-skilled technicians are assigned to customer support full time. In most cases, a problem can be solved over the phone in a few minutes. If not, repairs can be completed within 24 hours from receipt of product. Centralized repair service is available through ADC's Service Center.

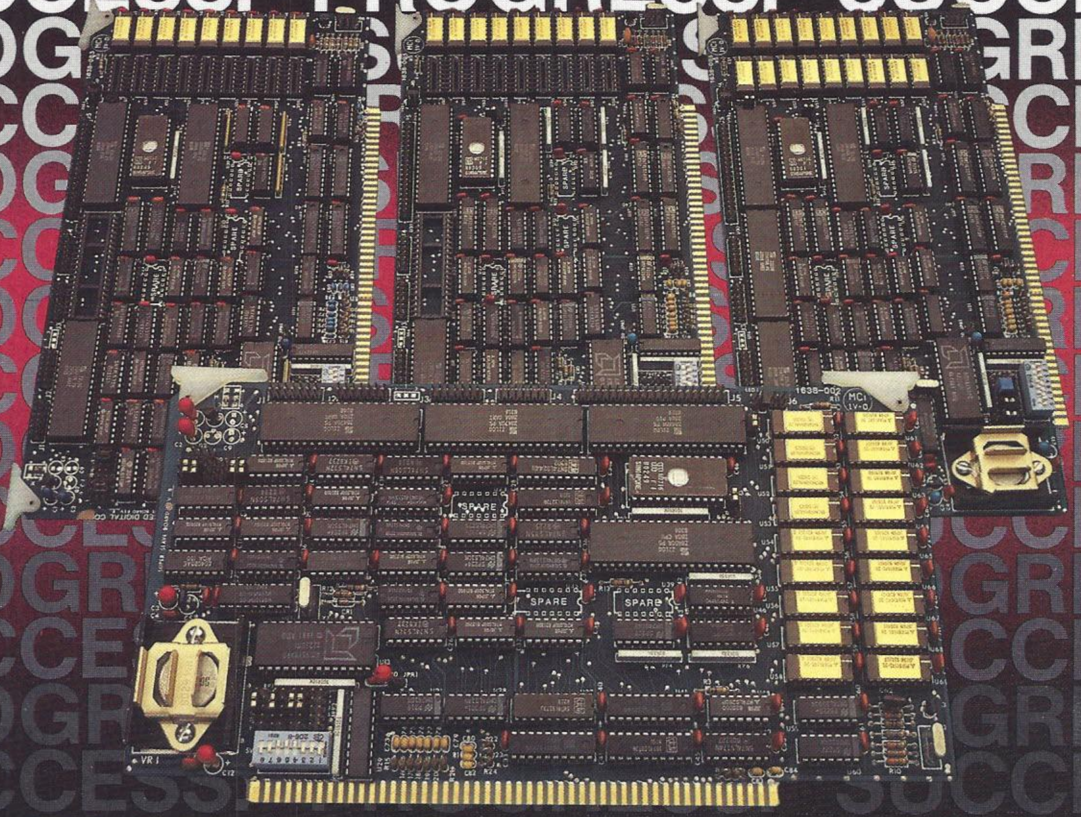
5. Market Penetration; Over 12,000 units installed.

With an installed base of over 12,000 units worldwide, ADC provides service to thousands of businesses and individual owners. By developing a superior line of S-100 products, ADC has become a major force in the industry. Customers include: US Navy, NASA, Harris Systems, Aerospace Corp.

6. Resources... Technical Expertise, Advanced Engineering.

ADC has assembled a team of highly experienced, industry renown specialists in engineering, manufacturing, marketing, sales and administration. ADC is a company of pioneers and innovators that have led the way and set the S-100 standards.

PROGRESS. SUCCESS. PROGRESS.
SUCCESS. PROGRESS. SUCCESS.



Founded in 1980, by a team of highly skilled and experienced engineers, Advanced Digital Corporation has grown to become a leading supplier of Single-Board Computers, Multi-User, Multi-Processor computer systems and memory subsystems for the S-100 Bus and PC Compatible Products marketplace. We're a company that has made tremendous growth without sacrificing the major factor that led to it. Quality...quality in our product, in our people, and in the way we do business.

Leading our success in the S-100 market, is a legacy of products that represent the needs of the S-100 market. Products such as Super System II, Super 16, Super 186 and Multi-Slave have made ADC the success it is today.

SUCCESS STORIES:

- SUPER SYSTEM II...A fully integrated system that runs under CP/M 2.2® for single-user applications or TurboDOS® for multi-user installations for both 8 and or 16-Bit including networking to PC's.
- SUPER 16...The cost effective 16-Bit 8 MHz Multi-User Single Board Slave processor.
- SUPER 186... The first high speed 8 MHz 16-Bit S-100 Single Board Computer.
- MULTI-SLAVE...A three user board contains three independent computers at 8 MHz and runs the TurboDOS operating system.

Progress. Success. It's the Story of our Company.

The success of ADC in the S-100 market is also due to the following line of quality products: • Super Star • Super Quad • Super Six • Super Slave • Master 9.

At ADC we're an aggressive company, a progressive company, an innovative company. To be successful, we know that each product we develop must be totally unique to the market.

Our progress and success is with products that have gained the highest ratings and the respect of the entire S-100 industry. Our success is based on superior technology, unique features, and the most cost competitive prices in the market. When you're looking for progress, look to ADC, because it's the story of our company.

ADC

ADVANCED DIGITAL CORPORATION

ADVANCED DIGITAL CORPORATION
5432 Production Drive • Huntington Beach, CA 92649
(714) 891-4004 • (800) 251-1801 • Telex 183210 ADVANCED HTBH

ADVANCED DIGITAL U.K. LTD.
27 Princes Street, Hanover Square • London W1R8NQ-United Kingdom
(01) 409-0077 • (01) 409-3351 • TLX 265840 FINEST

® TurbosDOS is a trademark of Software 2000. ® CP/M 2.2 is a trademark of Digital Research Corporation.

SEE US AT FALL/COMDEX 86 • BOOTH 1480

BYPASSING THE OBSOLESCENCE GAME

The September 1 issue of *InfoWorld* announces on its first page: 'IBM Planning to Replace PC Line, Analyst Predicts.' The September 8 issue of *Newsweek* carries a 16-page full-color ad from IBM promoting the PC line and accessories.

We know that it will happen again because it always has. Computers arrive and they soon depart. And companies will advertise their obsolete micros until the eve of the day they announce a new replacement (sometimes the replacement is not even obsolete!). And, until the eve of that day, there is no lack of takers who wake up the next morning to find that their $2495 system is now advertised in *The Morning News* for $995.

For the nontakers there are alternatives. S-100 systems continue to succeed as the never-obsolete micro. Consider that the S-100 bus was first introduced more than 11 years ago. Probably 90% of all the types of other computers that were born since are now long dead, never to be spoken of again. With them died most publications that over the years have emerged to support specific machines. *S-100 Journal* was introduced only 1 year ago. That's more than 10 years after the S-100 bus came on the scene. Yet, our magazine is flourishing and still steady on its course to support only S-100 computers.

During this past year, more than 30 new S-100 boards have been introduced. These are performance products that continue to expand the versatility and capabilities of the bus. One-megabyte S-100 static RAM boards are now available from CompuPro, Lomas Data Products, and other companies. A two-megabyte static RAM board is available from MACROTECH International. From Intelligent Computer Designs comes the HD64180-based single board computer that carries on-board everything from memory to LAN controller. InterContinental Microsystems places four HD64180 chips on a single board allowing four users on a single card. Inner Access is shipping SCSI interface boards that allow S-100 systems to run optical disk drives. Z80 slave boards have allowed running 8-bit software on 16-bit systems. Now Advanced Digital lets you have three Z80H CPUs running at 8 MHz on one slave board and throws in six serial ports. Teletek, Earth, ICM, and other S-100 companies are implementing sophisticated LANs that connect S-100's to many other computers and allow as many as 4,000 workstations. Finally, to top a successful year, we now have 32-bit S-100 boards from Cromemco.

The list goes on. It will continue growing in the future. In the most fluid industry of the century, only the S-100 bus is permanent this side of the VAX. If you own an S-100 system, keep it. If you don't, go out and buy one.

Jay Vilhena

S-100 JOURNAL ADVERTISEMENT POLICY

S-100 Journal is fully dedicated to the support of the S-100 bus. This editorial policy extends to the advertisements that are accepted for publication.

We welcome advertisers of S-100 systems, S-100 boards and other components, and most products or services adequate for the typical S-100 environment or which we judge to be of interest to the S-100 community in general. This includes terminals, printers, modems, software, accessories, supplies, and other products or services. Advertisements of nonS-100 systems and their accessories are not published. Advertisements of MS/PC-DOS software are usually not accepted. However, they can be accepted if the software is specially configured or produced for an S-100 system, or if it is also available, and is being advertised, for other operating systems.

Ads that are primarily S-100 and which include some nonS-100 material are usually accepted but are subject to our final approval.

We reserve the right not to publish any advertisement and to decide whether or not a product or service meets our S-100 criteria.

The Peak of Performance

SCALE THE HEIGHTS OF PRODUCTIVITY

Sure, you've proven that in your hands a computer is a productive tool. But if you haven't teamed up with a SemiDisk you have heights yet to climb!

IT'S NO MERE RAMDISK

SemiDisk has been leading the way for Disk Emulators since their inception. If you've seen RAMdisks you know what it's like to load programs in an instant, and read or write files without delay. Unlike alternatives, the SemiDisk offers up to 8 megabytes of instant-access storage while leaving your computer's main memory free for what it does best - computing!

KEEP A GRIP ON DATA

Go ahead, turn off your computer. Take a vacation. With the battery backup option, your valuable data will be there in the morning even if you aren't. You'll sleep better knowing not even a 5 hour blackout will sabotage your files.

NEW LOWER SEMIDISK PRICES THAT WON'T SNOW YOU UNDER

	512K	2Mbyte
IBM PC, XT, AT	$495	$995
Epson QX-10	$595	$995
S-100, SemiDisk II	$799	$1295
S-100, SemiDisk I	$595	—
TRS-80 II, 12, 16	$695	$1295
Battery Backup Unit	$130	$130

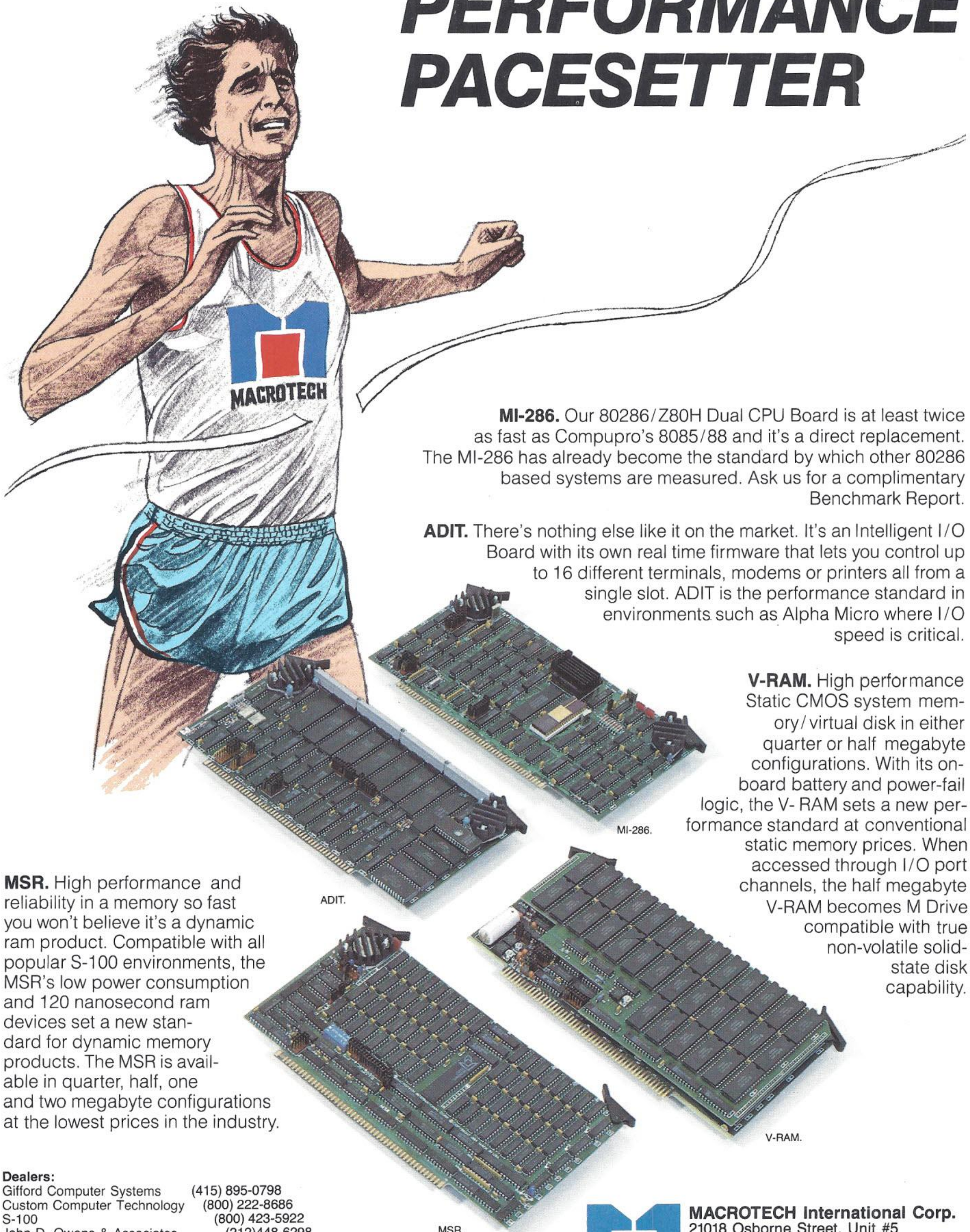
Software drivers available for CP/M 80, MS-DOS, ZDOS, TurboDOS, and VALDOCS 2.

SEMIDISK

SemiDisk Systems, Inc.
P.O. Box GG, Beaverton, Oregon 97075
503-626-3104

Call 503-646-5510 for CBBS/NW, and 503-649-8327 for CBBS/Aloha, all SemiDisk equipped computer bulletin boards, (300/1200/2400 baud) SemiDisk, SemiSpool trademarks of SemiDisk Systems.

MACROTECH—*STILL THE S-100 PERFORMANCE PACESETTER*



MI-286. Our 80286/Z80H Dual CPU Board is at least twice as fast as Compupro's 8085/88 and it's a direct replacement. The MI-286 has already become the standard by which other 80286 based systems are measured. Ask us for a complimentary Benchmark Report.

ADIT. There's nothing else like it on the market. It's an Intelligent I/O Board with its own real time firmware that lets you control up to 16 different terminals, modems or printers all from a single slot. ADIT is the performance standard in environments such as Alpha Micro where I/O speed is critical.

V-RAM. High performance Static CMOS system memory/virtual disk in either quarter or half megabyte configurations. With its on-board battery and power-fail logic, the V-RAM sets a new performance standard at conventional static memory prices. When accessed through I/O port channels, the half megabyte V-RAM becomes M Drive compatible with true non-volatile solid-state disk capability.

MSR. High performance and reliability in a memory so fast you won't believe it's a dynamic ram product. Compatible with all popular S-100 environments, the MSR's low power consumption and 120 nanosecond ram devices set a new standard for dynamic memory products. The MSR is available in quarter, half, one and two megabyte configurations at the lowest prices in the industry.

Dealers:

Gifford Computer Systems	(415) 895-0798
Custom Computer Technology	(800) 222-8686
S-100	(800) 423-5922
John D. Owens & Associates	(212)448-6298
In England; Fulcrum (Europe) Ltd.	(0621) 828763

Macrotech dealers also include most Compupro Systems Centers, Heathkit Electronic Centers and Alpha Micro Dealers.

MACROTECH International Corp.
21018 Osborne Street, Unit #5
Canoga Park, CA 91304
(800) 824-3181 • in Calif. (818) 700-1501
Telex: 9109970653

editor interface

Welcome to our biggest and best issue so far. We'll stabilize at this size now for a few issues just to catch our breath. Whenever we think that production is under control, we decide to increase the number of pages and off schedule we go again. No more! Expect the next issue exactly 3 months from now like a well-behaved quarterly must do.

I want to once more thank all our subscribers for the support you continue to give us. It is your enthusiasm about the S-100 bus that runs this magazine. The S-100 bus belongs to all of us, not to any particular company, and I hope you feel the same about S-100 Journal. To that end, I'll incorporate into the magazine as many of your ideas and suggestions as possible within our S-100 editorial direction.

Whenever you have questions, comments, suggestions, or complaints about the magazine, any of our articles, or any of the subjects discussed in this column, please write to Editor Interface, S-100 Journal, PO Box 1914, Orem, UT 84057. We prefer to receive your letters typewritten (dot matrixed?).

Let's read some of the mail since the last issue:

More PC Feelings

I recently received my first copy of *S-100 Journal* and was very pleasantly surprised by the content and editorial quality. I am extremely pleased to **at last** find a strictly S-100 oriented magazine.

It is certainly 'hiding your head in the sand' to ignore IBM. On the other hand, I don't own an IBM, I don't want one, and I am frankly tired of IBM. Congratulations on publishing ***S-100 Journal***!

Mark Pickerill
Salinas, California

I just read the Editor Interface section in issue number 3 and I'd like to address a few of the points raised.

It seems there are lots of people out there who, for one reason or another, would have the major manufacturers of S-100 boards and systems refocus their attention to the task of building S-100 versions of the IBM PC. While I can certainly sympathize with the frustrations of those who wish they could run all those 'Gee Whiz!' PC graphics programs on their S-100 computers, I don't expect to see the S-100 market go in that direction.

My advice to those who would like to run all those great programs written for the PC is to go straight out and buy a PC. The little devils are so cheap now that they can be bought for very little more than the price of a standard computer terminal. They can also be networked to S-100 systems, thus providing the best of both worlds. And they are so universally supported by peripheral vendors and software developers that the IBM-oriented computer hobbyist can stay busy for years just finding out about all the stuff.

Now, before you jump to the conclusion that I have somehow lost all of my marbles and am advocating the abandonment of the S-100 bus (which for an owner of a CompuPro System Center would seem a silly thing to do), let me point out a few areas in which the venerable IEEE-696 standard is, in my opinion, years ahead of the PC.

Modularity — the ability to upgrade to later technology as it becomes available. (Let's just wait and see how impressed the owners of all those 8088/8086 PC compatibles will be with the next generation of 'IBM-compatible' software products which will **only** run on 80286/80386 processors with multimegabyte directly-addressable RAM areas. Just smile as you tell them how you swapped-in that new CPU board and installed that new operating system.)

Multiuser Effectiveness — As the PC crowd continues to wait for a clear winner to emerge in the multiuser arena, and quick-fix approaches continue to proliferate as inexperienced PC dealers grapple with this 'new' technology, we S-100 users can enjoy the benefits of mature, multiuser/multitasking systems which have been working effectively for years.

Long-term Flexibility — To a growing business, in these times of rapid technological change, the ability to reconfigure, expand and enhance the central computer system can have a significant, positive effect on the bottom line. The modularity of an S-100 system can allow its owner to adapt the existing computer to fit changing requirements in cases where a less flexible system would have to be discarded and replaced.

Reliability — Most PC compatibles are designed by people who expect their machines to compete largely on the basis of price. The obvious consequence of that circumstance is that, during the design process, these products must often sacrifice a certain amount of 'reliability margin' in exchange for a higher return of performance for the consumer's dollar. This translates to a higher incidence of component failure than would be the case otherwise. S-100 systems, on the other hand, are for the most part being marketed to more sophisticated and demanding buyers who may place a higher value on system reliability. To many businesses, a loss of data or a few days of down time would represent a much greater expense (in terms of dollars) than the extra cost for a more reliable system.

For these and many other reasons, I am confident that the S-100 bus will continue to flourish in the

(continued on page 55)

696 bus

A FAST WAIT STATE GENERATOR

Don Pannell

696 Bus is a regular column that concentrates on the hardware aspects of the IEEE-696 bus (i.e., the S-100 bus) and answers questions that readers might have about the IEEE standard.

Don Pannell, our 696 bus columnist, is an S-100 hardware enthusiast and is coauthor of the IEEE-696 standard. He bought his first S-100 system (an Altair 8800b kit) in January 1978 and has since designed and built most of his present components. These include a terminal, EPROM programmer, serial/parallel I/O cards, two designs of DMA floppy-disk controller, and a 68000 coprocessor card. Don is president of Peak Electronics and is technical editor for S-100 Journal.

If you have questions about the S-100 bus and IEEE standard, write to Don Pannell, PO Box 700112, San Jose, CA 95170-0112. Your questions can range from architectural concerns to how to interface a specific device or function with the bus. In future issues, Don will incorporate answers to the most common questions.

a major advantage of the S-100 bus is its ability to accept various speeds of devices on the same bus at the same time. This feature allows a system to be upgraded one card at a time and has helped prolong the life of many systems. To make this possible, a slow device must be able to place the bus master into wait states. This is done with a wait-state-generator circuit.

At first glance, the need for a fast wait state generator is not apparent. In fact, it seems to be a contradiction of terms. Why worry about speed when the reason for a wait state generator is to slow accesses down? Simply put, the faster the processor the less time there is for the wait state circuit to do its job. If a board is expected to work with an 8 or 10-MHz bus clock, and it needs to insert wait states as well, its wait state generator must be fast.

WHAT ARE WAIT STATES?

Wait states are used to stretch the bus read strobe (pDBIN) or the bus write strobe (pWR*) to allow slow devices to communicate with a fast processor.

When wait states are not used, read and write strobes can be as small as 0.9 tCY (tCY is the bus-cycle time — tCY values for common clock speeds are shown in Table 1). This implies strobes of 90ns for a 10MHz processor. But many I/O devices, like serial chips and floppy-disk controllers, require more than 90ns, and often more than 150ns, for their read and write strobes. The only way to interface these chips to a fast bus is to use wait states. Each wait state will increase the width of the strobes by 1 tCY.

Figure 1 shows the fundamental timing relationships that occur during a bus cycle. It depicts a bus access with one wait state (BSW) inserted between bus states 2 (BS2) and 3 (BS3). This wait state was inserted because the bus ready line (RDY) was low during the rising edge of Φ (Phi) in bus state 2 (BS2). If RDY were low again during the rising edge of Φ in BSW, another wait state would have been inserted. This method allows for any number of wait states to be generated for any access.

FUNCTIONS OF A WAIT STATE GENERATOR

The job of a wait state generator is to perform the functions outlined in Table 2, page 10. This does not sound difficult until you try to make the circuit work at 10 MHz while keeping the bus loading within specifications.

There is another problem: the IEEE-696 standard is not complete in this area. No specifications are given for the maximum amount of time allowed between the address or status lines becoming stable and the wait state circuit asserting RDY. Calculating these values from the existing timing specifications results in numbers that are impossible to use.

It is very useful for a designer to know how much time is allowed for the wait state generator to determine if the addresses are within the desired range so that RDY can be pulled low. This specification, if it were present, would be called tARDY for 'delay Addresses stable to RDY stable.'

To calculate tARDY from the documented IEEE-696 specifications, one proceeds as follows (please refer to Figure 2 for the read-cycle timing diagram):

$tARDY = tAST^* + tST^*\Phi - tRDY\Phi$

where

$tAST^*$ = Address stable prior to pSTVAL* low during pSYNC high. IEEE specification is 70ns.

$tST^*\Phi$ = pSTVAL* low prior to Φ high during pSYNC. IEEE specification is 0ns.

$tRDY\Phi$ = Setup time RDY to Φ rising. IEEE specification is 70ns.

Therefore,

$tARDY = 70ns + 0ns - 70ns = 0ns!$

This means that the wait state generator must pull RDY low 0ns after the addresses are stable. Obviously, this is an impossible circuit to build.

The other missing specification should define how much time the wait-state circuit has to pull RDY low, counted from the time the status lines become stable. This would be called tSRDY, 'delay status stable to RDY stable.' Its equation is:

$tSRDY = tSST^* + tST^*\Phi - tRDY\Phi$

where

$tSST^*$ = Status stable prior to pSTVAL* low during pSYNC high. IEEE specification is 40ns.

$tST^*\Phi$ = pSTVAL* low prior to Φ high during pSYNC. IEEE specification is 0ns.

$tRDY\Phi$ = Setup time RDY to Φ rising. IEEE specification is 70ns.

With these values,

$tSRDY = 40ns + 0ns - 70ns = -30ns!$

This part of the circuit has an impossible job: it has to know what is going to happen before it does.

In the real world, tARDY and tSRDY must both be positive numbers. It is the responsibility of the CPU vendors to insure that adequate time is allotted for wait state generators to perform their function. And it is the job of the wait state generator designers to make the circuit as fast as possible.

THE FAST WAIT-STATE CIRCUIT

The top of Figure 3 (page 11) shows the first half of a very fast wait state generator for I/O devices. It performs functions 1, 2, and 3 (as listed in Table 2). An I/O-device wait state generator is shown for two reasons. First, it is the type of wait state generator most likely to be used. Second, it is the worst-case circuit due to the values that must be decoded from the status bus during I/O operations.

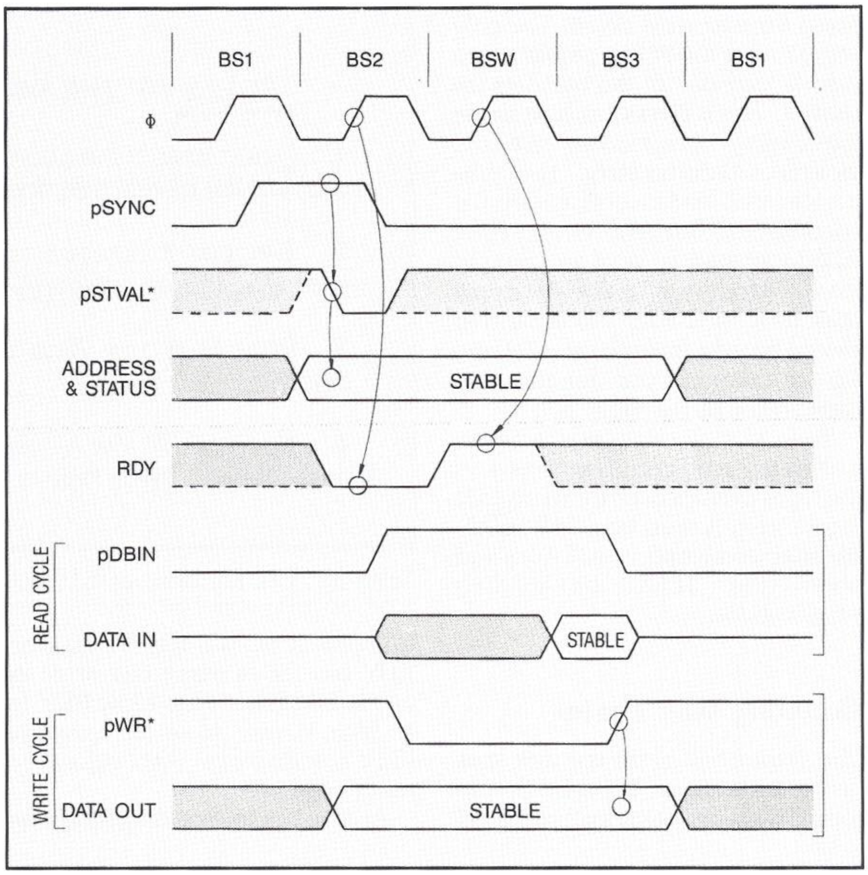


Figure 1. *Fundamental timing relationships of a bus cycle.*

This part of the circuit (Figure 3 — top) works as follows:

Chip C1 compares the bus address to that preset on the I/O board's dip switch. The output of C1 (pin 19) will go low whenever a match exists.

CLOCK	tCY	0.9 tCY
2 MHz	500 ns	450 ns
4 MHz	250 ns	225 ns
6 MHz	166 ns	150 ns
8 MHz	125 ns	112 ns
10 MHz	100 ns	90 ns

Table 1. *Bus clock speeds and corresponding tCY values.*

Gate A1 decides if the bus cycle is an I/O cycle by looking at the bus status lines. The output of A1 will go low whenever an I/O bus cycle is in progress.

Gate A2 combines the outputs of the above two functions. Its output will go high whenever the address and bus cycle are directed to this I/O card. Hence, this output is called BRDSEL for *board select.*

Gate B1 is used to pull RDY low if wait states are needed. This will happen whenever the board is selected (BRDSEL is high) and WAIT is high. WAIT comes from the second half of the wait state generator, and it will be discussed later.

Gate A3 is not part of the wait state generator. It is shown, however, because most designs need to know as soon as possible if a read or write operation is going to take place. An area where this may be important is in enabling the I/O board's internal data bus drivers. The only signals that can perform this function are the sOUT and sINP status lines. One or both could be buffered and then sent to gate A1. However, this would slow down the wait state circuit. The only way to solve the timing problem is to double-load one of the bus nets (as shown). This is allowed as long as the bus-loading specifications are not violated. Unfortunately, two LS-device input loads would violate the specification. But ALS works. (One ALS input load = ¼ of an LS input load). Therefore, chip A in the circuit must be a 74ALS02. The same chip should be used for both gates A1 and A3 to reduce the length and capacitance of the input net.

Table 3 lists the amount of time this wait state circuit takes depending on the types of chips used for B1 and C1 (again, chip A must be a 74ALS02). All listed combinations will work and meet the IEEE bus-loading specifications.

ROLES OF THE WAIT STATE GENERATOR

1. Determines if the address on the bus matches the address of the device that needs longer read or write strobes.

2. Decides if the bus cycle is of the proper type (for example, I/O or memory).

3. Pulls RDY low 70ns before the rising edge of Φ in Bus State 2.

4. Keeps RDY low for as many bus states as needed, and raises it 70ns before the next rising edge of Φ.

Table 2. *The functions of a Wait State Generator.*

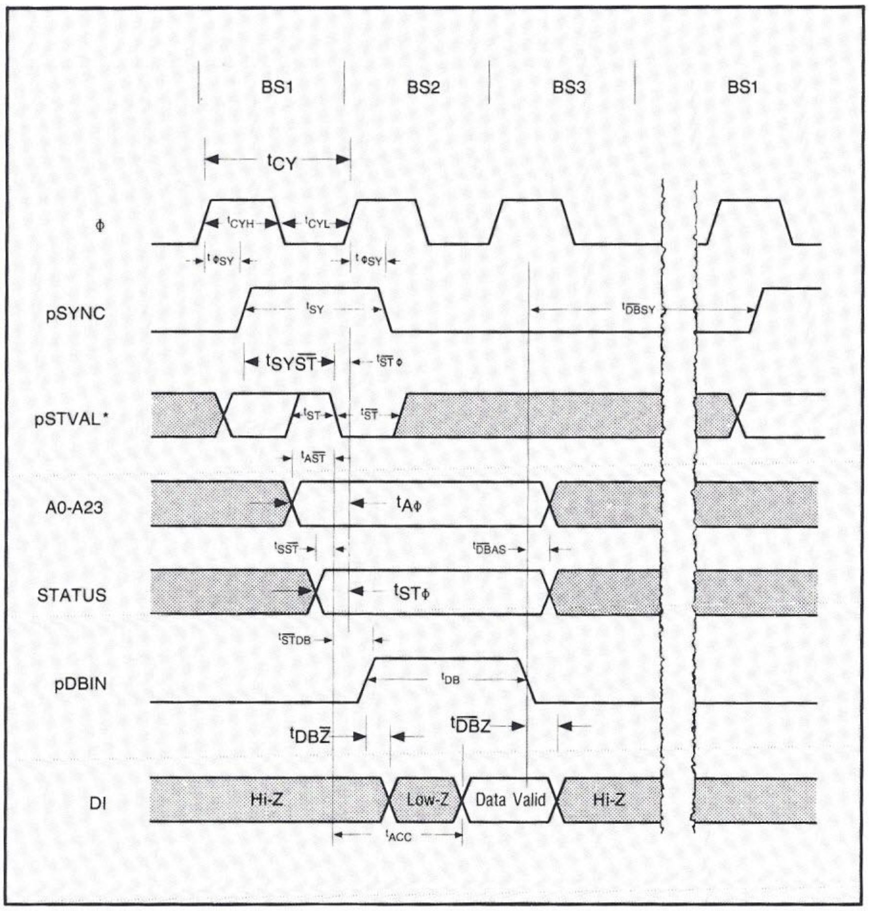


Figure 2. *Read-cycle timing diagram.*

Counting Wait States

The second half of the fast wait state generator is shown in Figure 3, bottom. This logic performs function 4 (as listed in Table 2) which is to keep RDY low for as many bus states as needed. When it does allow RDY to go high, it must do so (according to IEEE specifications) 70ns before the rising edge of Φ.

At slow bus speeds, these goals are easily met. But, if the bus clock is 10MHz (tCY=100ns), some design options are lost. At 2 or 4 MHz, the wait state counter (chip D1) can be reset by an inverted copy of pSYNC and clocked by an inverted copy of Φ (see Figure 2 for timing relationships). But, at 6 to 10 MHz, an inverted copy of Φ cannot be used. The rising edge of Φ must clock the wait state counter so that there is the maximum amount of time for the counter's outputs and RDY to change before the next rising edge of Φ. Table 4 lists the possible delays (from a rising edge of Φ until RDY is high) for the bottom circuit in Figure 3.

For any system, the maximum delay allowed from Φ to RDY is tCY− 70ns. Table 5 lists this value for common Φ values.

Tables 4 and 5 show that it is possible to meet the IEEE specification for the RDY setup time of 70ns, but, for a fast clock, it is difficult.

Since the number of wait states must be counted by the rising edge of Φ (as opposed to the falling edge of Φ), an inverted copy of pSYNC cannot be used to reset the counter. If it were used, it would be impossible to insert only one wait state. This is because the first rising edge of Φ that should clock the wait state counter would fail to do so. The counter would be held in reset mode by pSYNC still being high (pSYNC must be high on the rising edge of Φ in bus state 2 — refer to Figure 2).

To solve this problem, the wait state

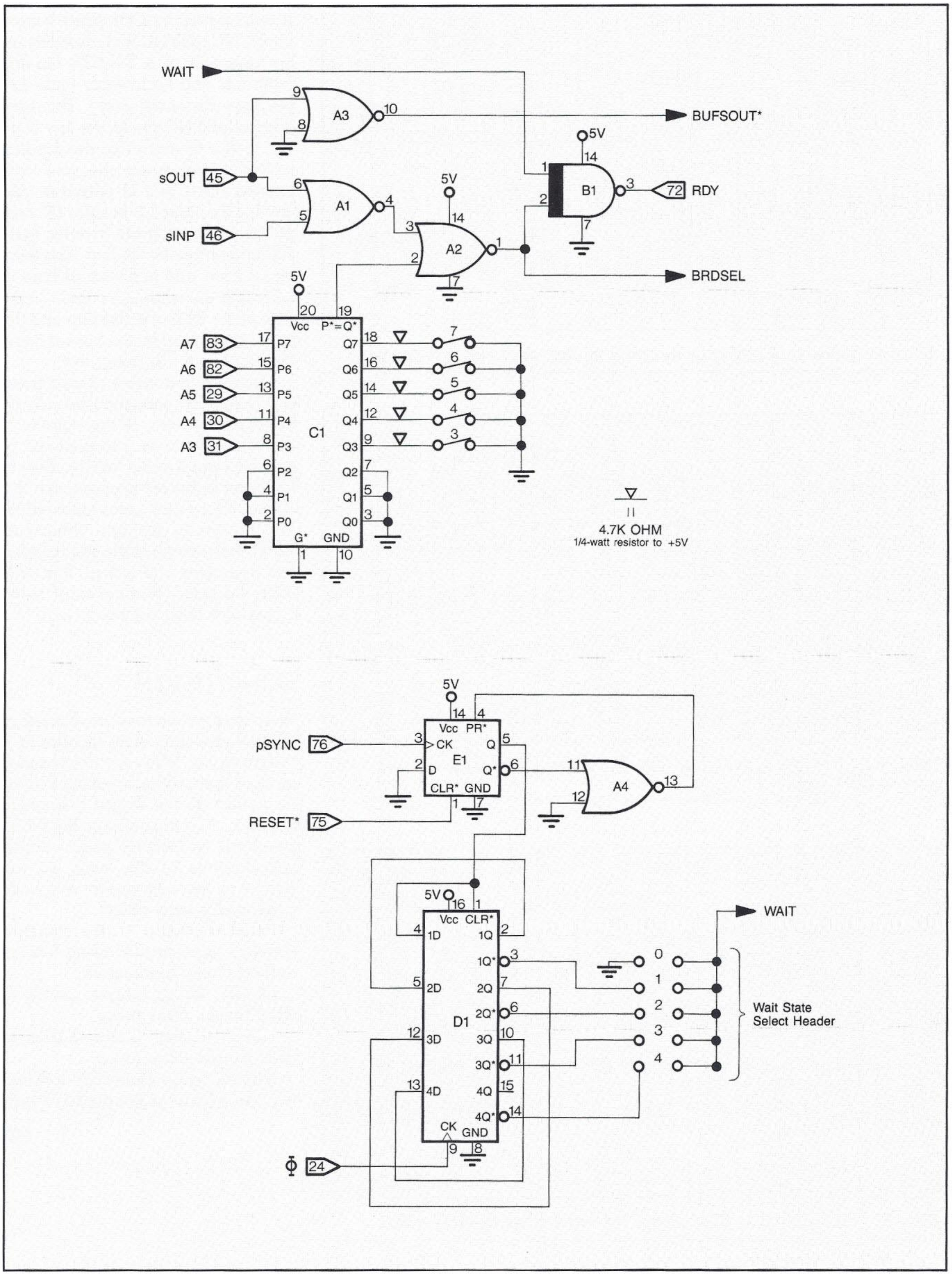

WAIT
A3
BUFSOUT*
sOUT 45
sINP 46
A1
A2
5V
B1
RDY 72
BRDSEL
C1
Vcc
P*=Q*
A7 83
A6 82
A5 29
A4 30
A3 31
P7
P6
P5
P4
P3
P2
P1
P0
Q7
Q6
Q5
Q4
Q3
Q2
Q1
Q0
G*
GND
4.7K OHM
1/4-watt resistor to +5V
pSYNC 76
RESET* 75
E1
Vcc PR*
CK
D
Q
Q*
CLR* GND
A4
D1
Vcc CLR*
1D
2D
3D
4D
1Q
1Q*
2Q
2Q*
3Q
3Q*
4Q
4Q*
CK GND
Φ 24
WAIT
Wait State
Select Header

Figure 3. *Circuits of the Fast Wait State Generator.*

CHIP B	tSRDY	tARDY	
		C1=LS688	C1=ALS688
B1=LS38	50ns	63ns	60ns
B1=ALS38	40ns	53ns	50ns
B1=S38	32ns	45ns	42ns

Table 3. *Worst-case timing delays from addresses stable to RDY stable.*

CHIP B	Φ to RDY Delay		
	D1=LS175	D1=ALS175	D1=S175
B1=LS38	57ns	36ns	49ns
B1=ALS38	58ns	37ns	50ns
B1=S38	35ns	14ns	27ns

Table 4. *Worst-case timing delays from the rising edge of the clock cycle to RDY high.*

CLOCK (Φ)	tCY	MAXIMUM Φ to RDY
2 MHz	500 ns	430 ns
4 MHz	250 ns	180 ns
6 MHz	166 ns	96 ns
8 MHz	125 ns	55 ns
10 MHz	100 ns	30 ns

Table 5. *Maximum Φ-to-RDY delays for various bus clock speeds.*

counter is reset by a 'chopped' version of pSYNC. pSYNC is chopped by using one half of a 74LS74 flip-flop (chip E1) and an inverter (gate A4, but any inverter will work). The rising edge of pSYNC clocks the low D input to the Q output of the flip-flop which in turn resets the wait state counter (D1). The Q output is held low for the time it takes the Q* output to go through the inverter (gate A4) and preset the flip-flop. This takes about 25ns and is plenty of time to fully reset the wait state counter (chip D1). Note: Both the flip-flop and the counter are reset by the bus reset line (RESET*, pin 75) going low.

The desired number of wait states is selected by connecting (usually by jumper plugs) one of the outputs of the wait state counter to the input of gate B1 (signal name WAIT). Zero to four wait states are possible with this circuit. If zero wait states are needed, WAIT is always held low. Otherwise, WAIT will go high at the start of each bus operation and will go low only after the selected number of rising edges of Φ have occurred.

CONCLUSION

Wait state generators are necessary when connecting slow devices to a fast processor. While the devices may be slow, the wait state generator must be as fast as the fastest bus speed allowed. Unfortunately, the IEEE-696 standard is lacking some timing specifications in this area, so the design of the high-speed wait state generator is very critical.

Future revisions of the standard should specify the following new or revised timing parameters:

tARDY, Delay Address Stable to RDY Stable, 65ns (new).

tSRDY, Delay Status Stable to RDY Stable, 55ns (new).

tRDYΦ, Setup Time RDY to Φ Rising, 20ns (current standard is 70ns).

d/OS REVISITED

Gary Feierbach

Last time we compared three similar operating systems available for the Motorola 68000: d/OS, AMOS/L, and Mirage. Among them, d/OS is emerging very strongly due to the tireless work of Mike Lewis and his associates at d/Soft. We will now take a closer look at this operating system.

SCREEN FEATURES

Several aspects of d/OS set it apart from other multiuser operating systems on microcomputers. Most apparent is the 'look' of the operating system. Both d/OS and its utilities make extensive use of CRT features such as cursor positioning, reverse video, intensity control, etc. Many commands and utilities have status displays or are menu-driven with block cursors. This is not just for aesthetics but to provide information and easy-to-use options not normally available in system utilities.

Most second-party operating systems assume a teletype-like terminal since they cannot count on any particular terminal being attached to a given port. In d/OS all commands that control terminal functions are generic. The terminal driver then translates the generic commands into those required by the terminal in use. And d/OS allows a given terminal driver to be loaded at any time. If a user logs in with a terminal that is not setup for that port, then a simple command (e.g., DRIVER WYSE50) loads a new terminal driver for that port.

MODULARITY AND SYSTEM GENERATION

Device independence and modularity have been carried to a near limit and include terminal, character I/O, disk, tape, and CPU (clock and timer) drivers. Making new drivers is not terribly difficult, and source code for a number of sample drivers is provided.

This modularity makes it easy to configure the operating system for new combinations of disk and tape drives, terminals, and jobs. A menu-driven program is provided, making system generation even easier. This program allows the user to completely and accurately reconfigure the system and the job environment in as little as 30 seconds, assuming that the drivers already exist on the system. This feature is of particular importance to VARs who have to support many system configurations and have to generate new ones with relative ease. With other systems, many dealers try to limit the number of configurations they sell since system generation is typically difficult. With d/OS, Henry Ford's quote 'You can have any color you want as long as it is black' need not be a computer vendor's motto.

WANTED
OPERATING SYSTEMS EXPERTS

We are expanding the *Multiuser OS* column to include several authors. Each columnist will need to be expert in only one operating system. Gary Feierbach will continue to write about d/OS, Mirage, and AMOS/L, and we are looking for other columnists to write about TurboDOS, Concurrent DOS, THEOS, Unix, OS-9, S1, and other multiuser operating systems currently running on S-100 micros. Each columnist will also be expected to read mail and answer questions from readers.

Columnists must have a solid knowledge of the OS they write about and must be currently running or implementing it on S-100 systems. A columnist may be a vendor of the operating system but cannot be part of the software company that produces the operating system. If you are interested in becoming a columnist for *Multiuser OS,* please write to *S-100 Journal,* PO Box 1914, Orem, UT 84057, or call Jay Vilhena (801-373-0696).

MULTITASKING

A user can initiate any number of tasks with the LAUNCH command. Each task launched takes space from the system memory heap with its terminal output directed to an associated file. These tasks are spun off as temporary jobs that release the acquired heap space after completion. Utilities are available to examine the status of the jobs as they progress and to view the terminal output. A super programmer will never again have to hog three terminals to maintain high productivity. Programs can assemble and compile while the programmer edits or debugs other modules using only one terminal. Perhaps the programmer will loose some much-needed exercise by not jumping from one terminal to another.

THE SCHEDULER

The heart of any time-sharing system is the job (task) scheduler. Most microcomputer-based systems use a simple round-robin scheduler with fixed maximum time for each job. But d/OS sports a full-blown, dynamic-priority scheduler with enormous flexibility. Jobs can be assigned priority ranges within which the scheduler can operate. The scheduler promotes jobs that have been bypassed by higher-priority jobs until they finally get a piece of the pie. Some low-priority jobs may not get to run at all until very high-priority jobs are completed. This has a strong effect on the user's perception of the system response time, and it can dramatically change the number of users a given system can comfortably support.

WILDCARDS AND WILDFILES

All operating systems have wildcards like * (asterisk) that can be used in filenames, allowing a group of files to be dealt with collectively. Many commands (e.g., DIR, COPY, ERASE, PRINT, and BACKUP) use this facility. In d/OS, this concept is applied to an extreme that some would consider bizarre. Consider the following example:

```
*LS*,–*AB*,–*.LIT
```

This would specify the set of all files with LS in the filename, excluding all those with AB or with the extension LIT.

The next example, although more complex, also has a straightforward interpretation:

```
{A–F}@@#*.BAS,TEST.BAS,
TRIAL#.BAS,– C*.BAS,–TRIAL3.BAS
```

This represents the set of all files beginning with a letter from A through F, followed by two more letters, then a number, then any other characters, and the extension BAS. Also included are the files TEST.BAS and TRIAL0.BAS through TRIAL9.BAS. From this set we exclude any file that begins with C and has the extension BAS, and we also exclude the file TRIAL3.BAS.

These specifications can easily become unwieldy, but we can place them in a file and make them multiline for readability, editability, and future use. Assuming the wildcard specifications are put into the file SPEC1.WLD, then preceeding the filename with an apostrophe ('SPEC1.WLD) indicates a wildcard-specification file to the using program. These in turn may be part of a specification list as in:

```
'SPECX.WLD,'OTHER.SPC,–WS#.BAS
```

This would specify the set of files resulting from merging the files SPECX.WLD and OTHER.SPC, but excluding the files defined in WS0.BAS through WS9.BAS.

At this point, one is moved to ask why such a capability is necessary. A simple example will help. Suppose a user wants to dump a directory onto floppy disk, but the floppy disk capacity is less than the contents of the directory. The following copy commands could be used to place the file in two disks:

```
COPY FLP0:={A–M}*.*
```

(for first floppy)

```
COPY FLP0:=*,–{A–M}*.*
```

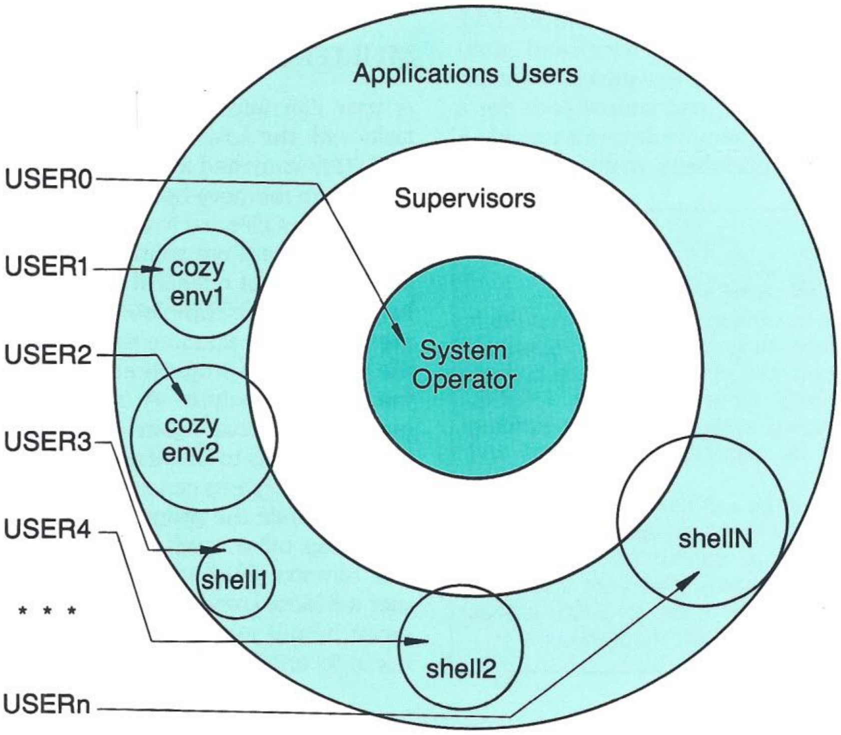
(for second floppy)

The user may utilize this wildcard capability in a program that he or she generates (e.g., PROG1) by using wildcard system calls or simply by typing:

```
X PROG1 wildspec
```

Here, X is a utility that will keep recalling the program PROG1 until all the files defined by 'wildspec' are used up. PROG1 will think it is called each time followed by a specification for a single file. With the X utility, the user can thus avoid the complexity of the wildcard system.

d/OS Security



SECURITY

When a user logs onto the system, he (or she) provides his initials, password, and company or department name or initials. At this point, he is granted access privileges that depend on his security level and the account, program, or shell he may be autologged into.

There are two programs, d/SHELL and COZY, that can keep users in a restricted enviroment. d/SHELL is a sophisticated menu-generating program that uses a block-structured script to generate a hierarchy of menus. COZY allows a user to talk to the system in the normal fashion, but under the control of a file that contains the names of the programs he is allowed to run.

TIMING AND BILLING

Each user is assigned an account number (typically associated with a company or department name) and a subaccount number (usually associated with the individual). Each time a user logs onto and off the system, the elapsed time is recorded for that account. Charges can be levied for three levels of CPU and three levels of disk-block usage. Charges are setup by account number, so rates can be different for different accounts. A billing program is provided, complete with source, so that it may be modified to handle a particular customer's requirements.

UTILITIES, UTILITIES

The operating system comes with a number of quality utilities, and each deserves some mention. Included is a macroassembler, Extended BASIC compiler, BASIC debugger, editor (for both program writing and word processing), text formatter for laser printer, disk caching, printer spooling, electronic mail, bulletin board, communications package, and of course all the usual utilities for disk maintenance, copying, and for poking around in the system.

Extended BASIC

The Extended BASIC compiler is AMOS/L-compatible at both the source and object level. It is also compatible with Softworks' BASIC compilers for the Macintosh, Atari ST, and Amiga computers. The compiler features implicit data-type conversion, allowing strings and numeric quantities to be freely mixed in expressions. Routines like FIX and VAL are still provided for compatibility and type forcing.

The most powerful feature of this BASIC is the MAP feature (borrowed from Alpha Micro). It permits the construction of very complex data structures that can involve arrays of strings, floating-point numbers, and integers from one to five bytes long. MAP statements may share the same area of memory. This last feature makes them particularly suitable for constructing record formats.

The BASIC debugger features a single-step capability that allows statement-by-statement execution, under cursor control, through source code. Break points can be set at particular statements or when variables meet a certain condition or change in value. Variables can be displayed at any time. BASIC programs can be checked quickly and thoroughly using this software. It makes the debugging technique of sprinkling PRINT statements throughout a program appear primitive and laborious.

Editor

The d/VUE editor, included with d/OS, was originally intended as a source code editor, but, over time, capabilities have been added that make it a reasonable word processor for letters and manuals. Editor functions include the ability to move blocks of text, set margins and tabs, and all the usual cursor controls to move quickly through text. For programmers, some useful features are auto-capitalization, command macros and macro-learning mode. Also included are search-and-replace (with or without query) and word-wrap functions. The commands are fairly intuitive, making it easy to learn.

Electronic Mail and Bulletin Board

Electronic mail is borrowed from large time-sharing systems and in d/OS it offers a few small refinements. Each user has a mail box, and mail can be sent from user to user, to groups of users, or to everyone on the system. In addition, general purpose mailboxes can be defined that are treated as bulletin boards by groups of users or by the system at large. Commands are available to sort, select, and read only messages on topics of interest.

In 'chat' mode, users logged on simultaneously can exchange messages in real time. With 'certified' mail, reply messages are automatically transmitted when the original message is read. A d/VUE-like editor

MTC

The Multiuser Software Company

SOFTWARE SYSTEMS

$

Business Financial Management

- General Ledger
- Accounts Receivable
- Accounts Payable
- Payroll
- Inventory Control
- Point of Sale (POS)
- Purchase Order Entry
- Customer and Vendor Analysis
- Mailing List Manager
- Sales Order Entry
- Sales Analysis
- Multiple Sets of Books as Standard Option
- Complete Financial Statements
- Fully Integrated - No Reposting
- Custom Modules Available

Doctor

Dental Office Management System

- Patient Check-Out with Recall Scheduling
- Automatically Print Insurance Forms
- Extensive Practice Analysis and Reporting
- Aged Accounts Receivable
- From Small Practices to Large Clinics
- Fully Integrated with Business/Financial Management System
- Automatically or Individually Prepare Bills/Statements
- Patient Tracking for Recall, Cancellations, Referrals, 'Work in Process,' etc.
- Enhancements for Marketing Practice/Services
- Mailing List — Mail Merge — Mass Mailings
- Contract/Budget Accounts

MTC Software Operates on all S-100 Systems under TurboDOS

(also IBM-PC and compatibles)

Dealer Inquiries Invited

MultiComputer Technology
Corporation
126 Northpoint, Suite 152
Houston, Texas 77060
(713) 847-0333

TurboDOS is a trademark of Software 2000
IBM-PC is a trademark of IBM

within the mail subsystem facilitates message composition. It even includes the macro-learning capability.

Communications

Two d/OS systems can talk to one another via telephone (or direct wire for that matter) using the communications software provided with the system. This software allows a user at one terminal to converse, through another port, with another system and to send files back and forth. In the 'transparent' or terminal mode, it appears as though one is sitting at a terminal on the remote system. In file transfer mode, two programs FSEND and FRCV are started at each end. These programs check each block transmitted, correcting errors when possible, or retransmitting the block if necessary. The size of the block transmitted is adaptive and dependent on line conditions.

APPLICATIONS SOFTWARE

Applications for d/OS include the large body of software already developed for the AMOS/L operating system. These include accounting, inventory control, database management, professional (medical, dental, legal, and CPA), restaurant and hotel management, finance, wholesale distribution, education, construction, and engineering packages. A number of database application generators are also available, including Andi (from d/Soft) and SUMUS (from CShare).

NEGATIVE ASPECTS

In d/OS there are a few shortcomings that hopefully will disappear soon. The most glaring is its adherence to the Intel (and DEC) byte order within a word. This stems from Alpha Micro's use of the Western Digital chip set before going to the 68000 processor. To be compatible, d/OS followed suit. On a system without DMA, one can simply reverse UDS and LDS on the 68K processor. DMA requires a more extensive modification. Currently, d/OS runs on S-100 systems from Inner Access Corporation (Intel or Motorola order) and Alpha Microsystems (Intel order). However, d/Soft plans to make a piggyback board that will plug into a 68000 socket and perform the byte swap for other manufacturers. At some point, a version of d/OS which will adhere to the Motorola byte order is likely. Mike Lewis, of d/Soft, indicated that such a version might be a year away (sooner if a manufacturer is interested in underwriting the changes.)

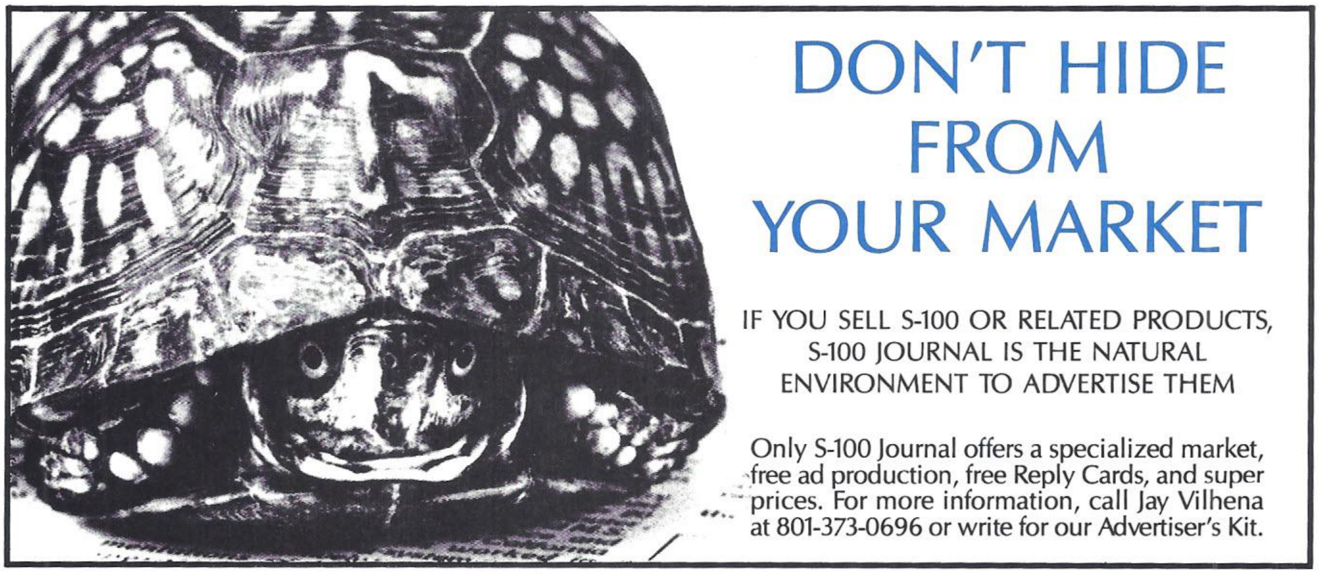
Another shortcoming is the use of 16-bit file pointers. This limits logical drive and therefore file size to 32 megabytes. A 90Mb drive is necessarily subdivided in smaller 30Mb logical drives. This has not been a serious problem, but now, with the advent of gigabyte CD-ROMs, it is beginning to look more limiting. Plans for changing this are in the works and are likely to be accomplished within the next six to nine months.

CONCLUSION

Presently, d/OS is going through a rapid evolution with a number of very dramatic improvements in just the last few months. If this pace continues unabated, d/OS could become the most outstanding operating system for microcomputers whether single or multiuser. It's a sound operating system with a good set of development tools. It makes efficient use of the 68000, leaving over 98% of the processor's computational power for jobs. The next d/OS revision level will take advantage of the performance improvements offered by the Motorola 68010 and 68020 processors. The current price of d/OS is $1250.

The production of good operating-system software requires clear, elegant thinking and accurate programming; d/Soft seems to have this talent.

d/OS is a product of d/Soft. For more information on d/OS contact d/Soft, 310 Cedar Lane, Teaneck, NJ 07666.


DON'T HIDE FROM YOUR MARKET

IF YOU SELL S-100 OR RELATED PRODUCTS, S-100 JOURNAL IS THE NATURAL ENVIRONMENT TO ADVERTISE THEM

Only S-100 Journal offers a specialized market, free ad production, free Reply Cards, and super prices. For more information, call Jay Vilhena at 801-373-0696 or write for our Advertiser's Kit.

MS-DOS 2.0 DEVICE DRIVER FOR THE HB-4 SERIAL INTERFACE S-100 BOARD

Text starts on page 38

THIS SOFTWARE WAS WRITTEN BY KEVIN PARKER. COPYRIGHT © 1986 BY KEVIN PARKER.

PERMISSION GRANTED TO REPRODUCE THESE PROGRAMS FOR INDIVIDUAL OR IN-HOUSE USE ONLY. THE SALE OF THESE PROGRAMS IN WHOLE OR IN PART WITHOUT WRITTEN PERMISSION FROM THE AUTHOR IS PROHIBITED.

```
;       This device driver is written for an HB-4 Serial Port card that has
;       all four 8250 UARTs installed.  Code is provided for polled driven and
;       interrupt driven I/O.  Each UART's serial line characteristics can be
;       read and manipulated through MS-DOS using Function 44h (I/O Control for
;       Devices).  Interrupt driven I/O buffer status can also be controlled and
;       read with this operating system call.  The format of the control strings
;       that will perform these functions is as follows:

;           For IOCTL Input (to get device information) - Up to 7 bytes are
;               returned that contain the following information:
;
;               1st word - Number of bytes waiting in the input buffer.
;               2nd word - Number of bytes in the output buffer.
;               3rd word - Baud rate in bits/sec (binary number).  The baud
;                          rate must be between 4 and 56K baud.
;               7th byte - Configuration information as follows:
;                          MSB                                          LSB
;                          ______________________________________________
;                          |           |           |                     |
;                          |_____|_____|_____|_____|_____|_____|_____|___|

;                            Stop bits    Parity     Data length (in bits)
;                            00 - 1 bit   00 - none   0101 - 5 data bits
;                            01 - 1 1/2   01 - even   0110 - 6 data bits
;                            10 - 2 bits  10 - none   0111 - 7 data bits
;                                         11 - odd    1000 - 8 data bits

;           For IOCTL Output (to set device characteristics) - A string (maximum
;               length <65536) that is composed of the following commands may be
;               sent to the device (parameters inside < > are ASCII characters
;               except for 'word' which is a binary number):

;               <B><word>   => Sets baudrate to 'word' (in bits/sec)
;               <FI>        => Flush input buffer
;               <FO>        => Flush output buffer (input buffer if polled I/O)
;               <PE>        => Sets even parity
;               <PO>        => Sets odd parity
;               <PN>        => Sets no parity
;               <D5>        => Sets data length to 5 bits
;               <D6>        => Sets data length to 6 bits
;               <D7>        => Sets data length to 7 bits
;               <D8>        => Sets data length to 8 bits
;               <S1>        => Sets 1 stop bit
;               <S2>        => Sets 2 stop bits

;       Set the base address of the board here.  Also select interrupt driven
;       I/O if you have the 8259 controller installed on the HB-4 board and
;       a master 8259 on another board (such as the SCP CPU Support Card).

                BASE            EQU  0000H
                INTERRUPT_IO    EQU   1
```

```
;       This device driver provides an interrupt-driven input and output buffer
;       for each serial port if selected.  If you set INTERRUPT_IO to 1, select
;       the size of the input and output buffers below:

INQ_SIZE        EQU     100   . ;Interrupt driven input buffer size (bytes).
OUTQ_SIZE       EQU     200     ;Interrupt driven output buffer size (bytes).

;               ------------------


; For an 8250 clock frequency of 1.8432 Mhz:
CLOCK           EQU     11520
UPPER_WORD      =       1           ;UPPER_WORD = CLOCK_FREQ  / (16 * 65536)
LOWER_WORD      =       49664       ;LOWER_WORD = (CLOCK_FREQ / 16) MOD 65536
DATA_AVAIL      EQU     1
XMIT_RDY        EQU     20H

        IF   INTERRUPT_IO
MASTER_59_0     EQU     0F0H            ;Base port address of system master 8259
MASTER_59_1     EQU     0F1H
SLAVE_59_0      EQU     BASE + 07H
SLAVE_59_1      EQU     BASE + 0FH
        ENDIF                                                       ;


;----------------------->   Device driver headers   <----------------------

DEV_CODE        SEGMENT WORD PUBLIC 'CODE'
                ASSUME  CS:DEV_CODE,DS:DEV_CODE

                ORG     0

        ;Device headers (1 for each serial port)

PRN_DEV:        ;Header for the Printer
        DW      EPROM_DEV,DEV_CODE      ;Next device header
        DW      0C000H                  ;Normal character device with IOCTL
        DW      STRATEGY                ;Routine to pass Command Header pointer
        DW      PRN_INT                 ;Device driver entry point
        DB      "PRN     "              ;Name of this device

EPROM_DEV:      ;Header for the serial EPROM programmer
        DW      MODEM_DEV,DEV_CODE
        DW      0C000H
        DW      STRATEGY
        DW      EPROM_INT
        DB      "EPROMPRG"

MODEM_DEV:      ;Header for the MODEM
        DW      AUX1_DEV,DEV_CODE
        DW      0C000H
        DW      STRATEGY
        DW      MODEM_INT
        DB      "MODEM   "

AUX1_DEV:       ;Header for the remaining port (currently not dedicated)
        DW      -1,-1           ;Last device driver in this file
        DW      0C000H
        DW      STRATEGY
        DW      AUX1_INT
        DB      "AUX1    "
```

```
;----------------->   Device interrupt service routines   <----------------------

;       This section of code may only be used if the 8259 Priority Interrupt
;       Controller is installed.
;       The following routines require a valid stack that has at least 4 words.

            IF  INTERRUPT_IO
;       Return sequence used by all service routines.
RETURN          MACRO
                IN      AL,DX           ;Are there any more interrupts pending
                CMP     AL,1            ;  from this serial port?
                JNE     SEL_INT
                MOV     AL,20H          ;Send End of Interrupt command to the
                MOV     DX,SLAVE_59_0   ;  slave 8259 (Auto-EOI mode doesn't
                OUT     DX,AL           ;  work for a slave 8259).
                POP     SI
                POP     DX
                POP     AX
                POP     BX
                IRET
                ENDM

;       'Transmitter Holding register empty' interrupt
TRANS_PROC:     MOV     SI,CS:[BX].OQ_FRONT
                CMP     SI,CS:[BX].OQ_BACK
                JE      DISABLE
                CLD
                LODS    BYTE PTR CS:[SI]                 ;Output the next byte at the
                OUT     DX,AL                            ;  front of the output buffer.
                MOV     AX,BX
                ADD     AX,OQ_END
                CMP     SI,AX                            ;Let pointer wrap around to
                JNE     NO_WRAP1                         ;  front of input buffer if
                SUB     SI,OUTQ_SIZE                     ;  it falls outside the area.
NO_WRAP1:       MOV     CS:[BX].OQ_FRONT,SI
                INC     DX                               ;Point to the Interrupt
                INC     DX                               ;  Identification Register.
                RETURN

DISABLE:        INC     DX              ;Disable transmitter interrupts if there
                IN      AL,DX           ;  is no more data in the output buffer.
                AND     AL,1101B
                OUT     DX,AL
                INC     DX              ;Point to the Interrupt Identification Reg
                RETURN


;--->  Interrupt service routine entry points:

EPROM_INT_RTN:  PUSH    BX
                MOV     BX,OFFSET EPB
                JMP     SHORT INT_DISPATCH
MODEM_INT_RTN:  PUSH    BX
                MOV     BX,OFFSET MODB
                JMP     SHORT INT_DISPATCH
AUX1_INT_RTN:   PUSH    BX
                MOV     BX,OFFSET AUXB
                JMP     SHORT INT_DISPATCH
PRN_INT_RTN:    PUSH    BX
                MOV     BX,OFFSET PRNB
```

```
;       All interrupts use the following code to determine what type of interrupt
;       occurred.  BX is used to point to the appropriate serial port's I/O buffer
;       block.
INT_DISPATCH:   PUSH    AX
                PUSH    DX
                PUSH    SI
                MOV     DX,CS:[BX + BLEN].PORT_BASE
                INC     DX
                INC     DX              ;Point to the Interrupt Identification reg.
                IN      AL,DX           ;Poll the 8250 to determine what caused
      SEL_INT:  CBW                     ;  the interrupt.
                MOV     SI,AX
                DEC     DX                      ;Point to UART data register.
                DEC     DX
                JMP     CS:INT_TYPE_TABLE[SI]

;       4 different interrupts from each UART
INT_TYPE_TABLE  DW      MODEM_INT_PROC
                DW      TRANS_PROC
                DW      RCVD_DATA_PROC
                DW      RCV_ERROR_PROC


;       MODEM status interrupt
MODEM_INT_PROC: ADD     DX,6            ;Read the MODEM Status Register to clear
                IN      AL,DX           ;  the interrupt.
                SUB     DX,4            ;Point to the Interrupt Identification Reg
                RETURN


;       'Received data error' interrupt
RCV_ERROR_PROC: IN      AL,DX           ;Throw away the received data and read
                ADD     DX,5            ;  the Line Status Register in order to
                IN      AL,DX           ;  clear the 'received data' and 'receiver
                SUB     DX,3            ;  error' interrupts, point to the
                RETURN                  ;  Interrupt Identification Register.


;       'Received data available' interrupt
RCVD_DATA_PROC: MOV     SI,CS:[BX].IQ_BACK
                IN      AL,DX                   ;Get the data.
                MOV     CS:[SI],AL              ;Store at the back of the input
                INC     SI                      ;  buffer.
                MOV     AX,BX
                ADD     AX,IQ_END
                CMP     SI,AX                   ;Make the 'Buffer-back' pointer wrap
                JB      NO_WRAP3                ;  around to the front of the buffer
                SUB     SI,INQ_SIZE             ;  if beyond the allotted area.
     NO_WRAP3:  MOV     CS:[BX].IQ_BACK,SI
                CMP     CS:[BX].IQ_FRONT,SI
                JNE     NO_OVERRUN              ;Update buffer starting point if
                INC     SI                      ;   buffer overrun occurs.
                CMP     SI,AX
                JB      NO_WRAP4                ;Don't let buffer 'front' pointer
                SUB     SI,INQ_SIZE             ;   go beyond the allotted area.
      NO_WRAP4: MOV     CS:[BX].IQ_FRONT,SI
    NO_OVERRUN: INC     DX                      ;Point to the Interrupt
                INC     DX                      ;  Identification Register.
                RETURN

            ENDIF                                               ;
```

```
;---------------------->   Strategy Routine entry point   <----------------------

STRATEGY        PROC    FAR
                MOV     WORD PTR CS:REQ_HDR_PTR,BX      ;Pointer to the function
                MOV     WORD PTR CS:REQ_HDR_PTR[2],ES   ;  request header passed here.
                RET
STRATEGY        ENDP


;--------------->    Device driver function request dispatcher <---------------

COMMAND_DISPATCHER      PROC    FAR
AUX1_INT:       PUSH    BX                          ;Use BX to point to the driver's
                MOV     BX,OFFSET AUXB              ;  command/data table.
                JMP     SHORT SAVE_ALL
EPROM_INT:      PUSH    BX
                MOV     BX,OFFSET EPB
                JMP     SHORT SAVE_ALL
MODEM_INT:      PUSH    BX
                MOV     BX,OFFSET MODB
                JMP     SHORT SAVE_ALL
PRN_INT:        PUSH    BX
                MOV     BX,OFFSET PRNB

SAVE_ALL:       PUSH    AX
                PUSH    SI
                PUSH    CX
                PUSH    DX
                PUSH    DI
                PUSH    DS
                PUSH    ES
                CLD
                LDS     SI,CS:REQ_HDR_PTR       ; DS:SI points to the request header.
                MOV     CX,[SI].BYTE_COUNT             ;Amount of data passed in CX.
                LES     DI,DWORD PTR [SI].XFER_ADDR    ; ES:DI holds transfer address.
                MOV     SI,WORD PTR [SI].COM_CODE
                AND     SI,0FFH
                CMP     SI,12                   ;Is this a legal command?
                JBE     VALID_COM
                MOV     AX,8103H                ;Return 'Unknown command' error.
                XOR     CX,CX
                JMP     SHORT FILL_HDR

     VALID_COM: SHL     SI,1
                MOV     AX,CS
                MOV     DS,AX
                MOV     DX,[BX + BLEN].PORT_BASE   ;Base address of I/O port in DX
                CALL    [SI + BX + BLEN]           ;Perform the requested function.
      FILL_HDR: LDS     BX,CS:REQ_HDR_PTR
                MOV     [BX].COM_STAT,AX        ;Status returned in AX.
                SUB     [BX].BYTE_COUNT,CX      ;Return actual number of bytes xferred.
                POP     ES
                POP     DS
                POP     DI
                POP     DX
                POP     CX
                POP     SI
                POP     AX
                POP     BX
                RET
COMMAND_DISPATCHER      ENDP
```

Listing continues on page 55

THE COMPLETE TurboDOS
MENU SYSTEM

```
Company Name Goes Here
System Name Goes Here
MAIN UTILITY MENU                                   USER 20

     0.  Return to Main System Menu
     1.  Backup and Restore
     2.  Directory Utilities
     3.  File Utilities (Copy, Rename, Set, etc.)
     4.  Disk Utilities (Change, Format, etc.)
     5.  Communications
     6.  Printing and Spooling
     7.  Miscellaneous (FIFO, Monitor, etc.)
     8.  Set System Date/Time
     9.  Change User Areas

>>> $  Enter Your Selection
```

```
Company Name Goes Here
System Name Goes Here
FILE UTILITIES                                      USER 20

     0.  Return to Main Utilities Menu
     1.  Copy a File
     2.  Rename a File
     3.  Delete a File
     4.  Activate a Do File
     5.  Set File Attributes
     6.  Show File Attributes
     7.  Type a File
     8.  Create an Autoload File
     9.  Dump a File

>>> $  Enter Your Selection
```

```
                              COPY A FILE
SOURCE  DRIVE  (A/B/etc.):__            DESTINATION DRIVE (A/B/etc.):__
    SOURCE  USER  AREA:__                   DESTINATION  USER  AREA:__
    SOURCE  FILE  NAME:_______              DESTINATION FILE NAME:_______
                        OPTIONS  (A,B,C,E,N,R,X,Y):____
```

NOTES:

1. File names may include 'Wildcard characters' '*' and '?'.
   - — '?' indicates a position in the file name that can be any character.
   - — '*' indicates a group of positions which can be any character.
   - — Example: LTR*.W?B would select all file names which begin with 'LTR' and have a file type (characters after the period in the file name) which starts with 'W', ends in 'B', and has any character in between.
2. Options available are:
   - 'A' — Copy only files which have been changed since the last backup.
   - 'B' — Allow file to span volumes.
   - 'C' — Allow diskette change during copy.
   - 'E' — Delete source file after copy.
   - 'N' — If file name contains wildcards, do not confirm each file.
   - 'R' — Replace, copy if file already exists at destination.
   - 'X' — Copy only if file does not exist at destination.
   - 'Y' — If File name contains wildcards, confirm each file.

*

MENU DRIVEN

SCREEN ORIENTED UTILITIES

USER DEFINABLE APPLICATION MENUS

☆ **$59.95** ☆

FEATURES SUMMARY

- Will run on All TurboDOS* Systems.
- Performs All TurboDOS* System Commands.
- User Friendly — No Syntax to Learn.
- Automatically Formats and Executes Command Lines.
- Fast Loading — Fast Executing Machine Language.
- Can Utilize Terminal Graphics Character Sets.
- Utilities Integrated with Application Menus.
- Gets First-Time Users Off to an Easy, Fast Start.

MTC

MultiComputer Technology Corporation
126 Northpoint, Suite 152
Houston, Texas 77060
(713) 847-0333

*TurboDOS is a trademark of Software 2000.

LOCAL AREA NETWORKS UNDER TurboDOS

Akin Orhun

A local area network is a communications system, much like a telephone system, where any connected device can use it to send and receive information. The networks discussed here are used exclusively for data transfer, although the technology is now also available for carrying voice and video signals.

As the name implies, a local area network (LAN) is used to cover a relatively short distance. Usually, a local area network will be limited to a department, a single building, or a group of buildings within close proximity. However, if used properly with a multiuser operating system such as TurboDOS, a local area network can have a large number of user stations.

An important characteristic of local area networks is speed; they deliver data fast. A person passing and receiving data over a local area network ideally experiences the same kind of response time as if data were coming from a local machine. To get this kind of response time, most local area networks operate at 1 to 10 megabits per second.

A LAN is a system made from building blocks which can be added and shaped as needed. The basic components are:

A **Cable** that transmits messages from one device to another.

Network Interface Cards that go between each computer and the cable.

A **Central Mass Storage** device, or devices, where network-shared data exists.

One or more **System Servers** which make network data available to the requesting party or parties.

A **Network Operating System** that makes provisions for using the network. The operating system discussed in this article is **TurboDOS.**

ADVANTAGES AND DISADVANTAGES OF LANS

Main advantages of local area networks are their flexible architecture, sharing of system resources (mass storage, printers, etc.), common access to data and program files over the network, and reliability against total system failure. The main arguments in favor of LANs are:

1. The sharing of peripherals reduces the per-user cost. Sharing in a properly designed network improves the reliability of an entire system.

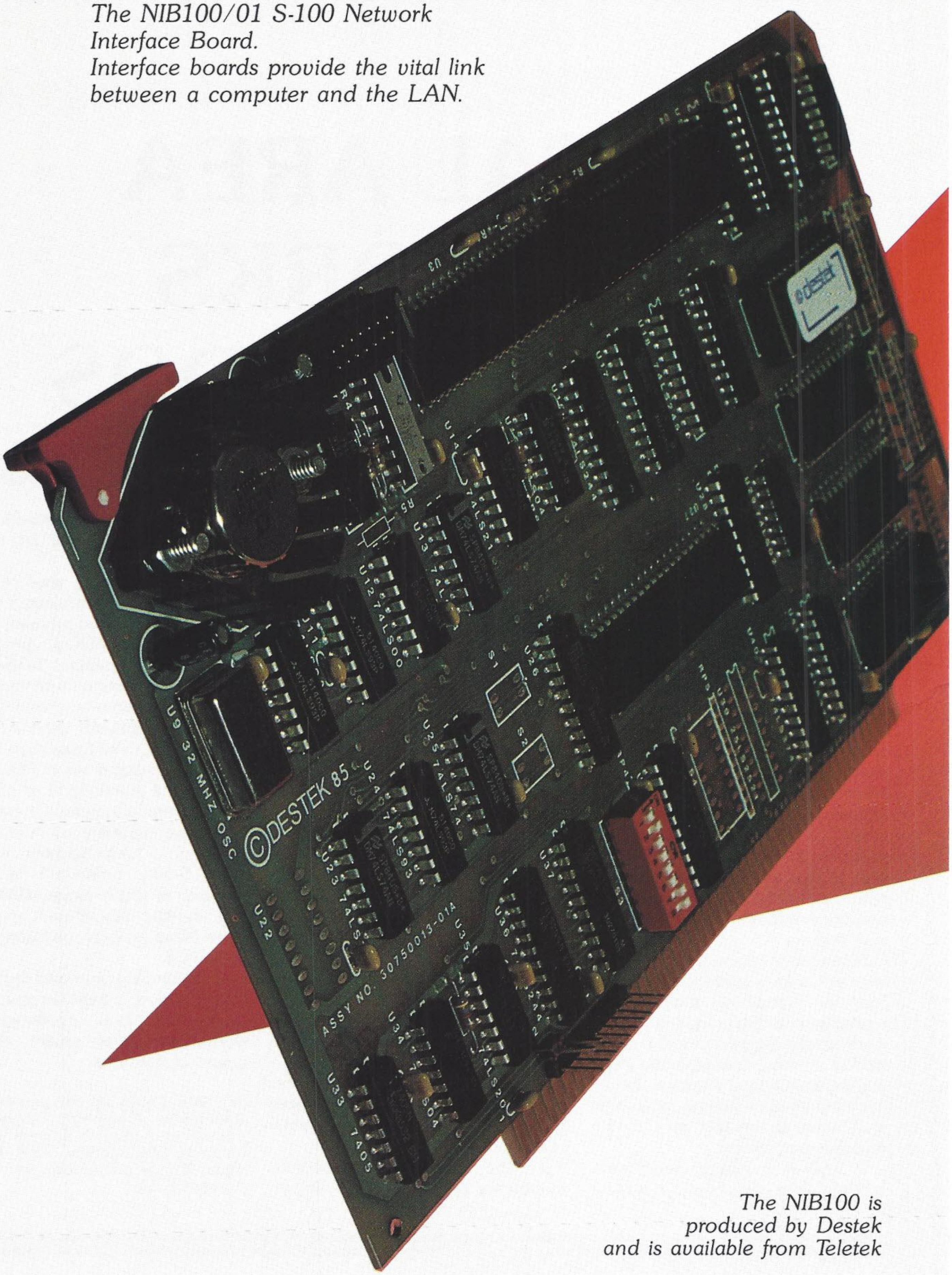
2. Better response time can be achieved through networking. Coupled with the distributed-processing approach of TurboDOS, properly designed and operated LANs are serious competitors to much more expensive mainframe approaches.

3. The peripherals attached to a network tend to be faster than those dedicated to stand-alone PCs. The cable speed (bandwidth) of all networks currently available far exceeds the speed capability of most PCs. However, a LAN between two or more S-100 TurboDOS systems operates at much higher efficiency due to the distributed master-server/slave-receiver philosophy of TurboDOS.

4. Networking promotes organizational efficiency, interaction and teamwork. Individual, nonintegrated workstations may create serious threats of data loss. ►

Dr. Akin Orhun is the president of TurboComp Computer Consultants and he is a research scientist at the University of California, Davis. In addition, Akin is a consultant for Teletek Enterprises, Inc.

The NIB100/01 S-100 Network Interface Board.
Interface boards provide the vital link between a computer and the LAN.



The NIB100 is produced by Destek and is available from Teletek

The ever-changing marketplace for microcomputers as well as advances in technology have made the local area network between multiuser, multiprocessor systems a strong alternative to timesharing on minicomputers or mainframes.

LANs, despite the many benefits they can bring to an office, can also be the source of certain problems. Even though LANs are logical extensions to personal computers, their operations are not similar. Software written for a single-user environment, although working in a LAN, may wreck data files if sharing files has not been resolved properly. Security problems exist because the cable, or one of the many workstations, may be easily available to outsiders. Or, a more common threat, authorized users may use their workstations to rummage through the files in the mass storage location. However, as it will be discussed, various safeguards exist in an operating system like TurboDOS.

STANDARDS

The computer industry cannot decide on the best, single solution for LANs. However, it can agree on one basic structure. Most LAN manufacturers have accepted the importance of an established structure and follow the scheme of the *International Standards Organization (ISO)*, referred to as the *Open System Interconnection* or **OSI model**.

The OSI model does not establish any particular standard. It is broad enough to include many standards, yet it provides a structure which requires enough similarities, so that network-to-network interfaces, called **gateways**, can be built and will enable the networks to communicate.

The OSI model defines the components and functions of a LAN in 7 layers, and sometimes an 8th layer, as shown in Figure 1. These layers are interdependent and the model is hierarchical. Each layer can only communicate with the layer below or above it.

Layer 1 defines the physical connection and is partly mechanical (connectors, cables) and partly electrical, specifying modulation techniques and voltages.

Layer 2 defines the formats used in the message units as well as the means of controlling access to the network. LANs break up messages into packets, or message units. Each packet carries with it its source, destination, and some type of error detection mechanism.

Layer 3 defines the switching and routing of information between network nodes. Network management, which includes the relay of status information and regulation of packet flow, is established in this layer.

Layer 4 defines the distribution of addresses on the network. If necessary, the procedure to divide the messages into smaller units is determined in this layer. Error detection and recovery are also handled here.

The layers 1, 2, 3, and 4 define the network and how it functions. The top three layers define how the network is used.

Layer 5, the session layer, defines the binding and unbinding of communication links, as well as the passage of data.

Layer 6 defines the translation of formats and syntax from an application to the network.

Layer 7 defines the support for applications run on the individual node points and specifies the manner in which applications can enter the network.

Some versions of the OSI model contain Layer 8 to cover the network management features. While the traditional 7 layers are stacked in their hierarchical configuration, the 8th layer runs vertically, interfacing with all 7 layers.

In a typical configuration of TurboDOS with Desnet (a networking system by Destek), the first two layers are handled by Desnet and the remainder layers by TurboDOS.

NETWORK TOPOLOGY

The physical layout of the network, its topology, is defined in the hardware layers. The cost and flexibility of the network installation are partly affected by topology, as is the system's reliability. There are four basic configurations for topology: the *Star*, the *Ring*, the *Distributed Bus*, and the *Token Bus*.

The **Star** topology (Figure 2) has a separate cable for each personal computer on the network. Each cable attaches to a central network processor. The Star is widely used in

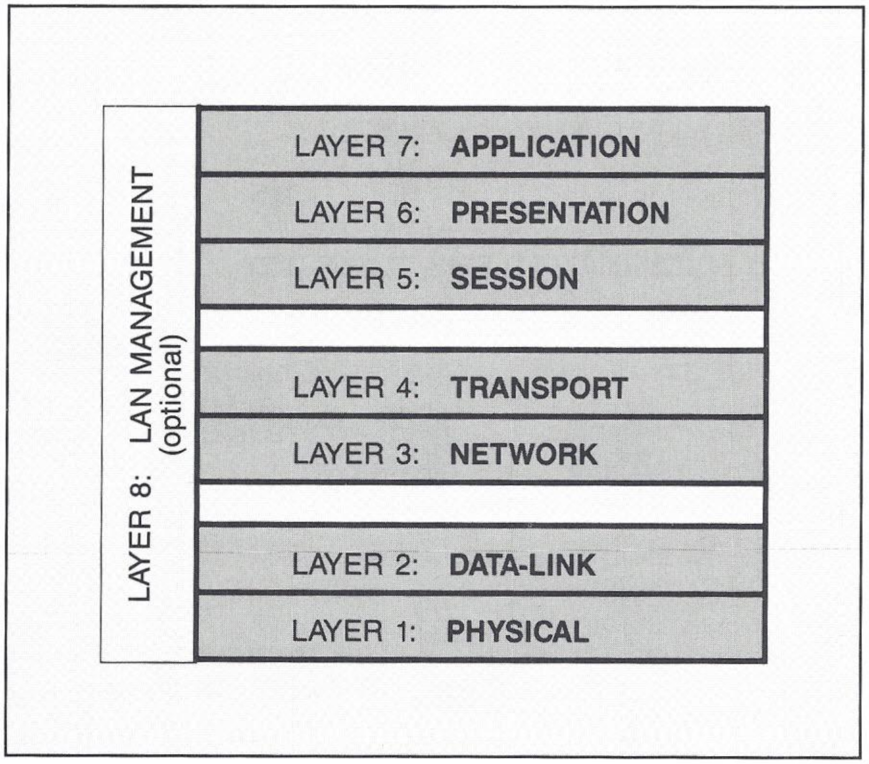


Figure 1. *Profile of the Open System Interconnection (OSI) model for LANs.*

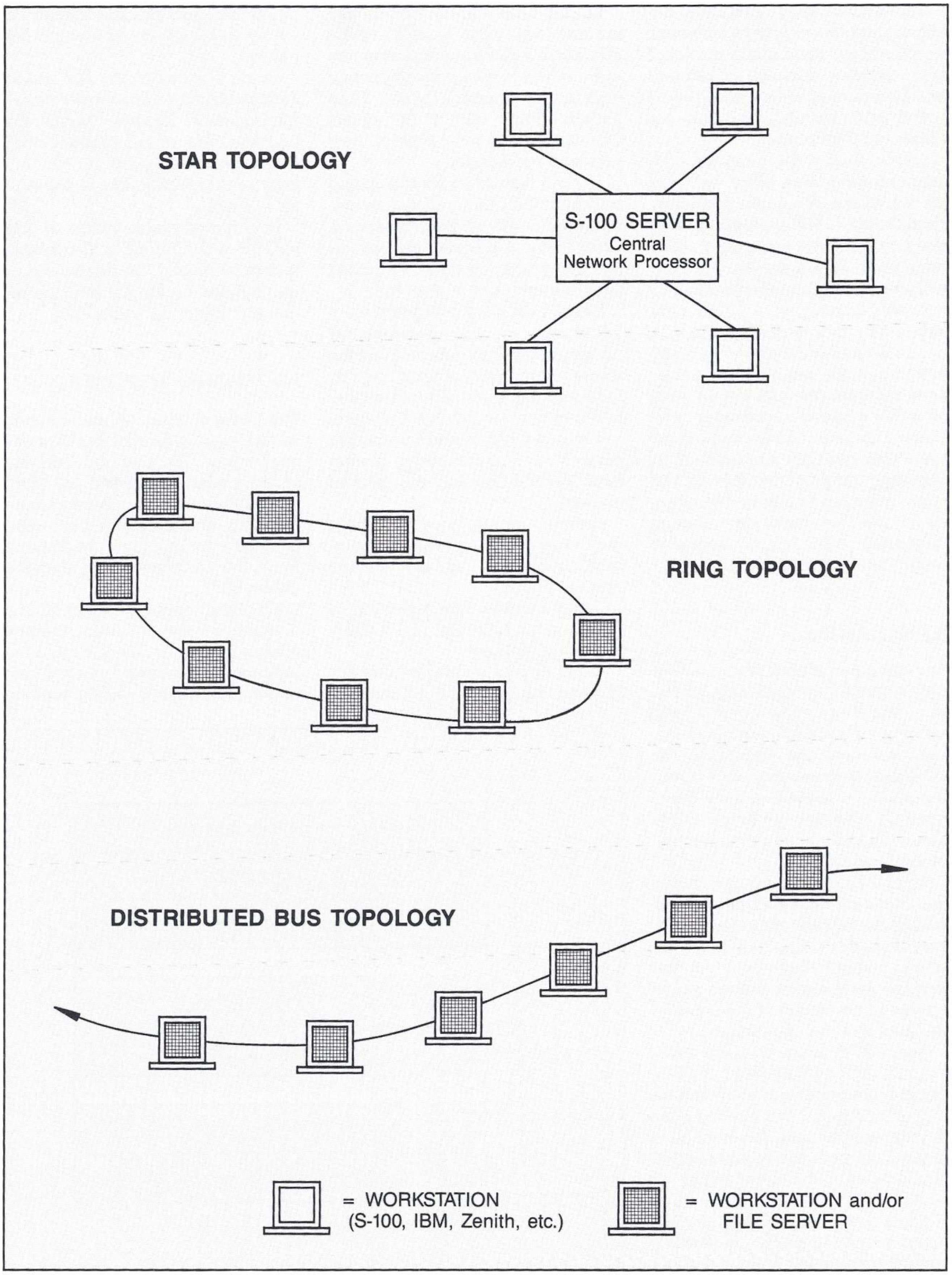

STAR TOPOLOGY
S-100 SERVER
Central
Network Processor
RING TOPOLOGY
DISTRIBUTED BUS TOPOLOGY
= WORKSTATION
(S-100, IBM, Zenith, etc.)
= WORKSTATION and/or
FILE SERVER

Figure 2. *Basic local area network topologies.*

host-to-terminal networks, PBX networks, and in a few LANs. The only advantages of the Star are the simplicity of its hardware and the possible use of existing phone lines as transmission media. On the other hand, it uses more cable than any other topology. It is also more cumbersome to install additional computers to the network since it requires a new cable each time. Another cause of concern is its vulnerability to the failure of the central processor.

The **Ring** topology (Figure 2) is a closed system. The cable passes through each peripheral and ends are closed to form a ring. The Ring may be the network configuration most subject to downtime. If a peripheral fails, the circuit is broken and the network operation is halted. The problem is corrected by running two parallel rings and connecting them so that a down machine or cable link can be bypassed.

A **Distributed Bus** (Figure 2) is the most widely used topology in local area networks. The bus is a single cable routed through the work area to which workstations can be attached at convenient points. This topology is highly reliable and flexible. Its only vulnerability is the failure of the cable.

The **Token Bus** topology (Figure 3), also called Distributed Star, is similar to the Distributed Bus. However, the Token Bus uses special interface devices called **HUB**s, which are attached to the bus at convenient points. Dedicated cables run from these HUBs to the networked systems.

Many combinations of the four basic topologies are possible. The *Star-Wired Ring* (Figure 4), for example, uses two Star networks in which the central processors are connected by two parallel rings, thus also providing backup redundancy.

The TurboDOS multiprocessor, multiuser system is itself a strongly bus-oriented system in which there is one master (server) processor and a multitude of slave processors (S-100 boards). Internally, the system resembles Star topology since the master server regulates the information flow on the S-100 bus. In networked TurboDOS systems, using hardware such as Destek's, there is a Token Bus topology amongst master server processors. Therefore, the final system can be referred to as *Token Bus Constellation* (see pages 30-31).

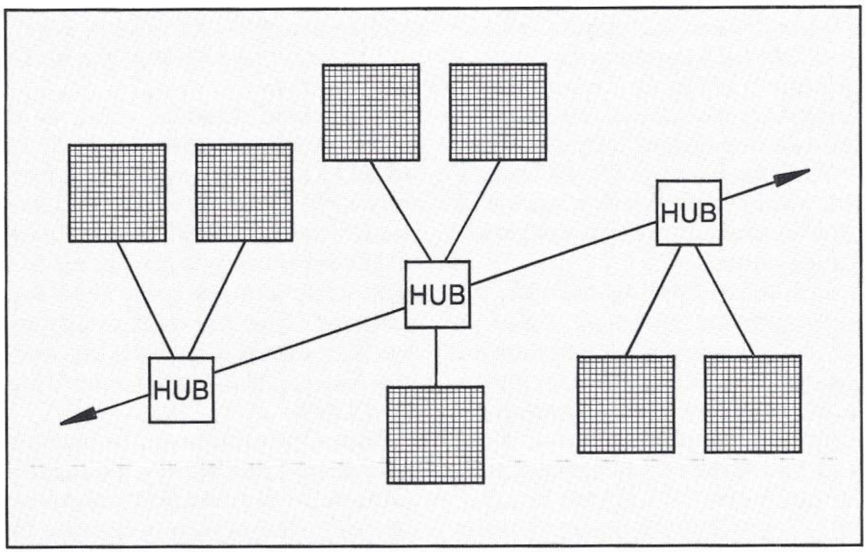


Figure 3. *LAN using Token Bus topology.*

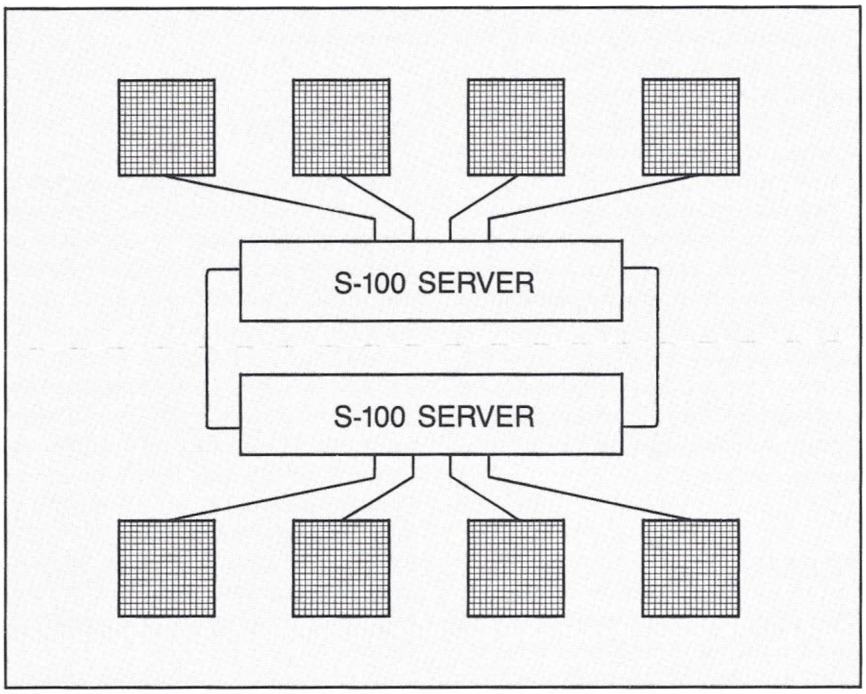


Figure 4. *LAN using Star-Wired Ring topology.*

NETWORK BUS ACCESS SCHEMES

Any topology which uses a common bus (i.e., Ring, Distributed Bus, or Token Bus) must have some means of regulating the access to the bus. Otherwise, nothing would prevent two workstations from making simultaneous transmissions and effectively blocking each other. Two access schemes are currently used: *Carrier Sense Multiple Access (CSMA)* and *Token Passing*. Access schemes are part of the hardware layers of the OSI model. Of the two major hardware layer protocols, CSMA is used by Ethernet (also by Desnet), and Token Passing is used by ARCnet.

In the **CSMA** method, also known as *contention scheme*, workstations must compete for an open time slot in order to access the bus. Most

CSMA schemes also include *Collision Detection (CD)* as part of their access method. While a workstation transmits, it continues to listen to the network to verify that the transmission is not being interrupted. If it detects any collision, it sends a jamming signal and retransmits after a predetermined interval.

In a **Token-Passing** network, an access-granting message, called a **token**, is generated during the system configuration and passed from station to station. Any station wishing to transmit must wait until a free token is offered. It then takes possession of the token and the network by altering a bit within the token. It then transmits. When done, it releases the token by resetting the bit to 'free' status.

Both schemes work well. Token Passing is defined as the *deterministic* access method, while CSMA is the *probabilistic* access method. During 'low traffic' or 'bursty' activity, the CSMA-CD method works better than the Token-Passing.

The two schemes are well established. Various protocols, such as ARCnet, Ethernet, and others have been transformed into simple chips from their complex protocols. None of these standards will disappear in the future. We expect to see sophisticated developments in internetting through gateways so that any one protocol can talk to another.

CABLES

The cable used in networking is specified at the hardware level. Factors considered by the manufacturers are: the cost of the cable, its transmission speed, maximum range of the network, susceptibility to electrical interference, and security threats.

The **Twisted Pair** is the least expensive of all alternatives. If multistranded and shielded, it can support data transmissions of up to 1 Mbits/sec. It fits well to Star topology and small department-level networks.

Baseband Coaxial cable is the choice of most LAN manufacturers. A single-channel cable, baseband uses electrical frequency signals and handles high-speed data rates of up to 10 Mbits/sec. Baseband coax is relatively inexpensive and free from interference. It is also easy to install and maintain. Desnet uses the baseband method.

Broadband Coaxial cable is a high-speed data medium which is similar to baseband. However, it carries a radio frequency signal which increases the bandwidth and permits multiple-channel capacity. The broadband cable is more costly than the baseband cable and requires expensive RF modems at each device interface. The broadband is not needed for most LANs.

Fiber-Optic cable is the newest technology. Light beam is modulated by the network to shape signals within the cable. This system is immune to electrical interference and outside interception. However, it is expensive, difficult to install and maintain, and very difficult to tap into for additional workstations.

OPERATING SYSTEM

Even though networking may seem transparent to the end user, there are immense differences in operating a single-user personal computer versus one that is a part of the network. Networking introduces new commands to control the computing system. It enables one station to communicate with another on the network. It may provide multitasking and multiuser features. From the user's point of view, the network operating system is the single most critical element of networking. Functionality, ease of use, management, data safety, and security are all features of the operating system.

A networking operating system supports applications on the network in much the same way that a local operating system supports applications on the personal computer. It has its own set of required utilities just like personal computer disk operating systems do.

TurboDOS and its support packages, such as TurboPlus, provide an extremely user-friendly atmosphere, file-locking mechanisms during simultaneous file accesses, security, electronic mail, and status information. TurboDOS also provides access to another operating system, like MS-DOS or PC-DOS, by running an emulator on its own bus-oriented 16-bit processors or by networking a PC to the bus. A PC networked to the bus may either operate under TurboDOS or it may be under MS-DOS and run TurboDOS/PC, a program which enables the PC to access the network. TurboDOS/PC is transparent to the user; it intercepts only the TurboDOS function calls and passes the rest to MS-DOS.

A utility is a special-purpose program integrated into the network. It can be thought of as the network operating system's own, dedicated applications software. Table 1 describes utilities that are generally found on TurboDOS' and other LANs.

There are two types of network servers available: the disk server and the file server. TurboDOS is a file server. In a networked TurboDOS environment there are multiple file servers that may be viewed as disk servers from a network point of view. Each one of these master servers is responsible for opening files and for locking files and records properly on its drives. The proper use of record and file locking is also the full responsibility of the application programs. If a single-user application program runs under a multiuser system, it should not be assumed that it automatically handles the file and record contention.

TurboDOS is truly a distributed-processing system in which master servers are also multitasking while serving the local processor demands. The operating system is designed such that most of the tasks are reentrant. Local servers (i.e., workstations) dedicate their resources mainly to the console activity and data processing. They delegate other tasks associated with the bus/network activity to the master servers. Therefore, the system performance on the multiuser and network level is outstanding.

APPLICATIONS SOFTWARE

Most network software used today is single-user, run by the 'grace' of the operating system. Depending on the specific operating system and application, this situation can cause considerable grief to the user. Just as the assumption 'if it works on floppy, it

LAN UTILITIES

Disk Serving permits a networked system to share mass storage devices. This aspect is handled flawlessly by TurboDOS with its disk buffers, multiple accesses, and so on. One multiuser TurboDOS system can support up to 16 logical drives. With proper design, this number is more than adequate to arrive at an arrangement of global and local disk drives for all users.

Print Serving allows user access to a common printer. Just like the disk drives, there are 16 logical printers available to each user, and they can be configured very easily within TurboDOS.

The **Print Spool** utility permits a user station to send data to a printer buffer, which is a file on the server. The spool file holds data until the printer becomes available. On networked TurboDOS systems, these spooled files can easily be left on any system in the network if the system is designed for it.

The **Login** utility implements the procedure for entering the network environment. TurboDOS provides an orderly login/logout procedure that allows common messages to all users to be displayed.

The **Password** utility provides the mechanism to protect data from unauthorized use. Beyond general-access protection, a password identifies a user profile. TurboDOS has two classes of users. Non-priviledged users may be locked into 2 or more disk drives and are not permitted to leave their own user level (1 out of 31 available). They may alter files in their own area as they wish. They may also access global files on a read-only basis if those files are placed on their disk drive or on a system disk drive. Privileged users have no restrictions. It is very easy for the network operator, who is a privileged user, to alter a user PROFILE under TurboDOS/TurboPlus.

The **File Specification** utility offers data protection similar to password priviledges. However, here restrictions are placed on the files instead of the user. TurboDOS does not use this mechanism.

Locking is a network-associated utility that temporarily reserves a file, record, or field for a particular user. This is necessary to avoid problems, or even destruction of data, when several users try to access the same file simultaneously. TurboDOS provides every locking mechanism currently available, including record locking.

Pipes are temporary volumes used to communicate between application programs running on different computers. Piped messages are stacked in a queue so that the first message in is also the first message out. TurboDOS provides pipes through disk FIFO buffers or much faster RAM buffers.

Electronic Mail allows memos, letters, and files to be sent from one workstation to another. TurboPLUS, the extension of TurboDOS, provides real-time message forwarding as well as electronic mail.

Configuration is another important tool that allows the user to issue commands to configure or reconfigure the environment dynamically. Currently, in networking TurboDOS, reconfiguration is a relatively simple and quick process, but it is not dynamic. MicroServe, the originator of TurboPlus, is currently working on this topic.

The **Network Monitor** is a diagnostic utility that allows a user to see activity on the network. Such utility exists under TurboPlus, and allows the user to get the snapshot or periodic update of the network activity.

Table 1. *Utilities typically found in local area networks, including LANs under TurboDOS.*

EXAMPLE OF A TurboDOS

BASED ON TELETEK S-10

The Local Area Network exemplified in this page is composed of five systems. Systems 1 and 2 are Teletek S-100 16-user systems. Each of these users has his/her own S-100 processor board which can be either a 8-bit or a 16-bit slave board. System 3 is the main server for the LAN. System 4 is a multitasking single-user S-100 system. Systems 5 and 6 are IBM PC, XT, AT, or compatibles.

Users in the network can perform different functions in the office, from data entry to programming, and each user can be configured to run under specialized hardware/software environments. One user, for example, may only require a 8-bit terminal for data entry, while another may need a special color monitor for graphics, a local printer, or real-world interface hardware.

Each of the six systems attaches to the network via a Destek NIB100/01 network interface board. Each LAN board is assigned a node number. The users on Systems 1 and 2 are also each assigned a unique number to facilitate communication.

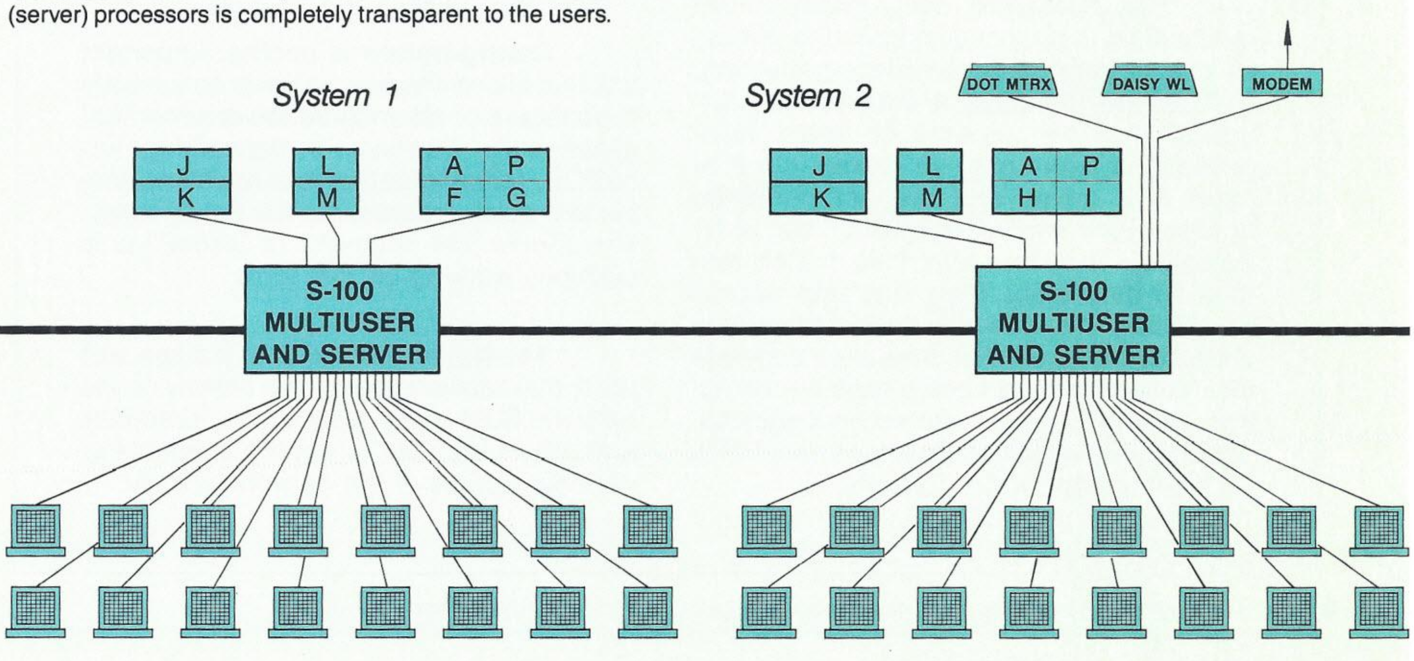
The master processors in systems 1, 2, and 3 are encharged with running the printers, manipulating the files, controlling queues, and performing other network tasks. Since each user station has its own dedicated (slave) processor, applications software can run very fast. One of the 8- or 16-bit slave processors can also be designated as a batch processor for other users in the network.

The disk drive assignment provides for local-system storage and global storage. Through easy-to-configure assignment tables, each user (including the users of Systems 4, 5, and 6) can be restricted to one or more disk drives and assigned to any number of printers in the network. A user can even be assigned to a command file which executes upon entry into the system, thus blocking that user from accessing other system resources and data.

The nonserver systems (4, 5, and 6) may run their local software and also have access to the LAN software. Local and global software may even run simultaneously. During accesses to disk drives, printers, batch queues, and other resources of the network, the assistance of the master (server) processors is completely transparent to the users.

OPERATING SYSTEMS RUNNING LOCALLY	
System 1	8/16-bit TurboDOS w/ TurboPlus
System 2	8/16-bit TurboDOS w/ TurboPlus
System 3	8-bit TurboDOS
System 4	ConcurrentDOS w/ TurboDOS/PC
System 5	MS-DOS/PC-DOS w/ TurboDOS/PC
System 6	MS-DOS/PC-DOS w/ TurboDOS/PC

TYPE OF SOFTWARE AVAILABLE TO SYSTEM	
System 1	TurboDOS, CP/M-80, CP/M-86, MP/M-80, MP/M-86, MS-DOS
System 2	Same as System 1
System 3	Same as System 1
System 4	TurboDOS, CP/M-80, CP/M-86, MP/M-80, MP/M-86, MS/PC-DOS
System 5	Same as System 4
System 6	Same as System 4



LOCAL AREA NETWORK

AND TurboNET COMPONENTS

MULTIUSER AND SERVER INTERNAL HARDWARE	
System 1 and System 2 (each)	1 master processor: Teletek Systemaster II (128K/8-MHz/Z80H) 8 8-bit slave CPUs: Teletek SBC-B (128K/6-MHz/Z80B) 8 16-bit slave CPUs: Teletek SBC-86 (512K/8-MHz/8086/87) 1 hard-disk controller: Teletek HD/CTC 1 LAN board: Destek NIB100/01
System 3	1 master processor: Teletek Systemaster II 1 8-bit slave CPU: Teletek SBC-B 1 hard-disk controller: Teletek HD/CTC (also controls tape) 1 Destek LAN board

DISK DRIVE ASSIGNMENTS

	TYPE	STATUS
A	¼ of local Hard Disk	LOCAL
B	8″ Floppy Drive	GLOBAL
C	5¼″ Hard Disk	GLOBAL
D	Tape Drive	GLOBAL
E	14″ Hard Disk (cartr.)	GLOBAL
F	¼ of 5¼″ Hard Disk	GLOBAL
G	¼ of 5¼″ Hard Disk	GLOBAL
H	¼ of 5¼″ Hard Disk	GLOBAL
I	¼ of 5¼″ Hard Disk	GLOBAL
J	8″ Floppy Drive	LOCAL
K	8″ Floppy Drive	LOCAL
L	5¼″ Floppy Drive	LOCAL
M	5¼″ Floppy Drive	LOCAL
P	¼ of 5¼″ Hard Disk	LOCAL

LAN-Oriented Commands of TurboDOS and TurboPlus

BB, BBDEL, and **BBLIST:** Used when designating a slave processor to handle background batch processing for other system users.

CHANGE: Used before removing disks from a remote drive.

DIRDUMP and **LOCATE:** These commands are for checking the user levels of the logical drives and to locate ambiguous files on the logical drives.

GO and **GONAME:** For assigning logical names to any of the logical drive user levels.

HELP: Explains TurboDOS and TurboPlus commands. Any HELP files may be added.

LOG: Enables user to record the progress of a job with time and date stamp.

LOGON and **LOGOFF:** For orderly entering and exiting the system.

MASTER: This command allows a priviledged user to attach a PC or slave processor console as a file-server (master) console.

PRINT: Allows the user to control the routing of print outputs.

PRINTER: Allows the user to control despooling to any selected printer.

PROFILE: Used by the network manager to alter the 'user ID/password' file structure and assign access status of each user.

REASSIGN: For reassigning the correspondence between logical and physical drives.

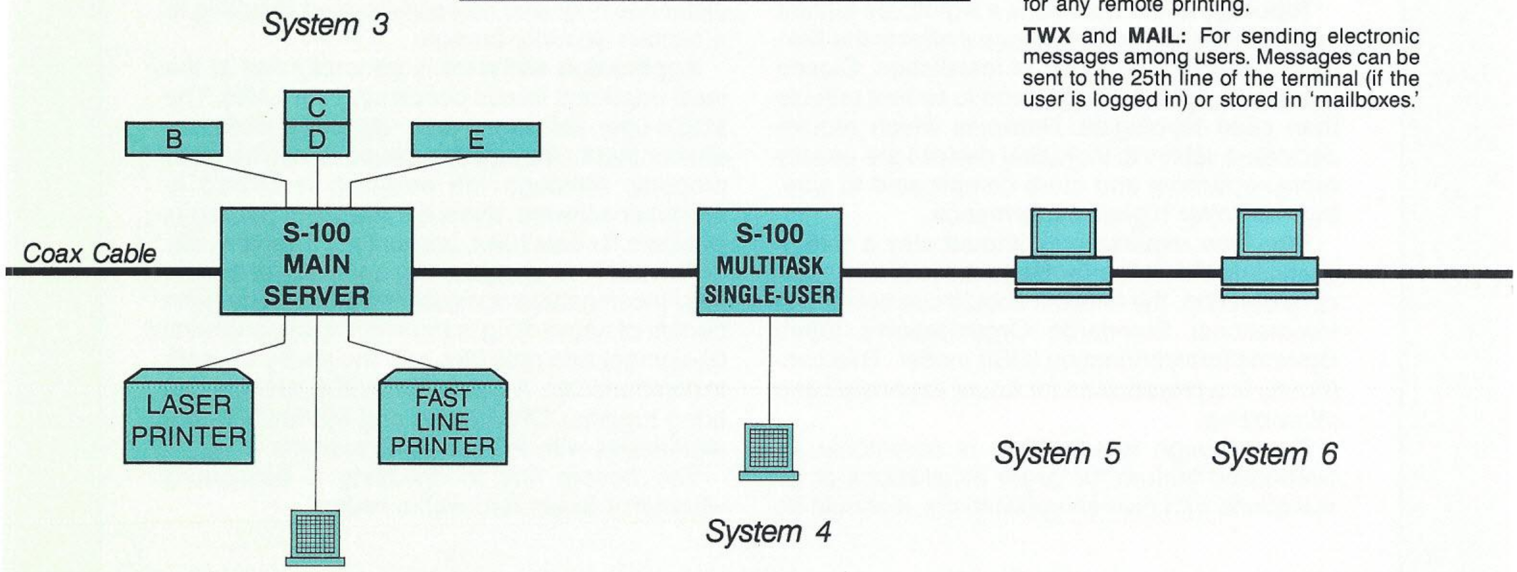
RESET: Closes any files that may have been left open by MS-DOS.

STATUS and **WHO:** These commands allow the user to inquire about the status and current activity on the network.

TURBOPC: Used in the installation of TurboDOS/PC. It allows the user to access remote disk drives and do remote printing over the network.

TURBOPRN: Used in the installation of the TurboDOS/PC 'print filter' which is required for any remote printing.

TWX and **MAIL:** For sending electronic messages among users. Messages can be sent to the 25th line of the terminal (if the user is logged in) or stored in 'mailboxes.'



GUIDELINES FOR SELECTING A NETWORK

Often, when network decisions are made, the technician's point of view prevails. From this perspective, many complex aspects of controversial technical issues must be resolved. But, from the manager/buyer's point of view, these technical matters obscure other important issues. Another alternative would be to ignore all technical aspects and make a choice on functionality alone. The best alternative, however, lies somewhere in between.

The first item to be discussed should be a **plan** for networking. This plan should state what an organization wants, how it will benefit, and what the benefits are worth. This plan should be the benchmark for comparing reality with expectation. The plan should be closely followed under most circumstances.

Access methods to the network may be another consideration. But, even though the methods discussed in the text (Token Passing and CSMA-CD) have advantages over each other under certain situations, their differences in affecting network performance are usually insignificant.

The **price/performance** relationship extends to local area networks. Generally, the more expensive the network, the better its performance. However, a typical TurboDOS local area network can cost 10 to 50 times less for a given performance when compared to a minicomputer or mainframe.

The **environment** in which a LAN is to be installed plays an important role in the overall plan. Electrical interference, heat, humidity, and distances involved may very well dictate the type of cable and connectors to be used for the task. The network's capabilities should not be exceeded in the proposed installation plan.

Topology of the network is a significant feature in the evaluation process since it affects the flexibility, reliability, and cost of installation. Closed topologies, such as Rings, tend to be less reliable than open topologies. Networks which require dedicated cables to individual devices are usually more expensive and more complicated to wire, but may offer higher performance.

Interface requirements should play a role in evaluating the networks. Regardless of any other consideration, the network should conform to the International Standards Organization's Open Systems Interconnection (OSI) model. This conformity is a prerequisite for future expansion and internetting.

Even though **internetting** is commonly an overlooked feature for larger installations or installations with diverse applications, it should be considered. Generally, network protocols are close enough so they can talk to each other directly, or they may require gateway devices and protocol converters. Gateways have a range of capabilities. The best gateway devices can pass a high percentage of application functionality along with the basic information.

Cable type helps determine both data rates and network range. Susceptibility to interference is wholly dependent on the cable. The overall network cost is also affected by the choice of the cable. Baseband coax cables are generally five times more expensive than the twisted pair, yet may offer the desired data rate and noise immunity.

For most networking applications, good response time depends upon various factors. The raw data rate over the cable, in the range of 1 to 10 Mbits/sec, is seldom the factor in overall speed. The amount of traffic on the network, processor speed, memory, architecture, and disk transfer speed are generally more important.

The **fault tolerance** of a LAN is extremely important. The additional cost of redundant components must be weighed against the potential loss due to network outage. LANs based on multiprocessor/multiuser TurboDOS systems with several file servers offer the desired redundancy and insurance against network outage due to a single component failure.

Data security is another important factor that should be considered. LANs have both interior and exterior security problems. The user id/password schemes may not keep undesirable parties away from sensitive files. Access to remote workstations is easier than in tightly controlled computer systems. Cables going over long distances may also be easily tapped resulting in a serious security breach.

Application software is generally one of the most important issues concerning the LANs. The single-user software, when used in a multiuser environment, may cause havoc if not handled properly. Although the selection is limited in multiuser software, there are adequate packages available for data base, accounting, inventory, etc.

The network should bring a **variety** of essentially incompatible computers together. The main benefit of networking is the sharing of peripheral equipment and data files and the ability of users to communicate. A typical TurboDOS network can bring together CP/M-80/86 and MP/M-80/86 environments with PC/MS-DOS systems.

The bottom line in choosing a network is whether it exists and works **today!**

should work on hard disk' is wrong, so is the assumption that 'single-user data-base-management software can be used in the shared environment of a local area network.'

Today, perhaps 99% of all local area networks are used for simple disk sharing. Since network programmers have made networks look like a single-user environment, less emphasis has been placed on the information sharing or communication capability of the network. Users are generally unaware of the potential they are missing. Similarly, due to a lack of understanding, users assume that sharing of communications and information is implicit. It is not! Network software provides tools for these operations. It is up to the application software to make use of these tools.

For example, a network operating system can provide locking down to the record level. But locking the record, deleting it, and unlocking it will not solve the problems that arise when several users want to manipulate the same database file. The only solution is to use a multiuser approach so that the multiuser software writes the file out to the disk and updates it properly.

There are three categories of software currently available. These are: *Unnetworked, Networked,* and *Multiaccess Networked.*

Software which abuses the operating system through some unorthodox scheme needs to be unnetworked form the system. This problem is relatively rare. Networked software is single-user software which runs on the networked system with the appearance that it is running on a single-user system. The third category of software, multiaccess, is aware that it is running on a multiuser environment and takes advantage of all (or most) of the operating system calls. Care should be taken while implementing single-user software, such as a data base management program, which has been upgraded to a multiuser environment. Generally, such an upgrade may solve problems associated with lock and unlock commands if only two people are using it. However, problems may arise under heavier usage. If heavy usage is a must, then a 'verbal semaphore' which schedules usage among network users must be arranged.

It is conceivable that if a DBMS (Data Base Management Software) package is ill designed and all of its elements are not updated, the reports written by one user may not reflect the current status of the data base. A major function of a multiuser DBMS is to keep track of who is doing what, and this does not have an equivalent on a single-user package.

Another useful feature for a multiuser data base is the capability of defining users. A password scheme that defines the access level of the user to the data base is an extremely desirable feature. TurboDOS and most other networks have the capability to define users, but not at DBMS level. This feature of the network operation system could be used as a prefilter to the DBMS' own password protection scheme.

Programs which run on memory most of the time, with infrequent access to the hard disk storage, obviously are better suited for network applications. Software requiring large amounts of disk access should be placed in a local storage medium so that the overall network load is reduced.

Common computer languages developed for single-user environments do not address themselves to the needs of multiuser and network requirements. However, multiuser versions of some of these languages are available. They do provide methods for orderly file retrieval and update in a multiuser environment.

ELECTRONIC MAIL

Electronic mail is probably the single most significant contributor to office productivity — even more important than spreadsheets and word processing.

Electronic mail is a message which is encoded as electrical impulses and passed over a transmission line. The message may be a memo, letter, file, graph, or any combination of these. Electronic mail eliminates the problems associated with getting a physical document to an intended party. Unlike telephone messages, both parties need not be available simultaneously. A computer-based message system (CBMS), such as those available on LANs, has many advantages over terminal-based electronic delivery systems.

In a TurboDOS system with the TurboPlus upgrade, it is possible to send real-time messages between users as well as traditional electronic mail. A message sent by one party is kept in mailboxes. Users have the option of deleting their mail after reading it. Electronic mail has its own editor. However, larger messages may be written by word processing packages, and the destination party may be informed about their existence and location through electronic mail.

Electronic mail and local area networks promote better communication among all employees of an organization thus helping to increase productivity.

ADMINISTRATION

The most practical approach to organizing a network is to appoint one person to manage it. Network managers should tailor the network so that each user 'sees' only as much of the network as it is necessary. The manager also educates users on the use of the network.

Initial planning of the network is absolutely necessary. Such planning determines which department gets what equipment, peripheral devices, and links to the common storage. Individual departments should access only those files needed. For example, shipping departments should not have access to confidential personnel records. Once this has been decided, individual users and their access status to the files should be determined. Certain files which are accessed by more than one department may be accessed by the users on read/write or read-only basis depending upon their needs.

Files stored on the hard disk can be given shared or nonshared status. Even though protection schemes exist, data on a common storage medium, such as a hard disk, is ultimately available to all users. In designing the network, care should be given in assigning local disk drives that are not accessed by all users. If floppy drives are available, sensitive files should be kept out of the networked system drives.

A suitable file-naming convention should be decided upon and used by all users on the network. Some network software should have the means to time-stamp files for further convenience. Also, a reliable backup system is a must for any network. Mirror-image backups should be avoided since this system copies bad and good sectors of the storage medium. Removable disk cartridges and file-by-file backup to a tape are two preferred methods. In general, a grandfather-father-son scheme with another backup stored at an 'off-site' is a desirable backup procedure.

A device log kept by the network manager is another useful administrative tool. This log should include schematics, printouts of diagnostics, dip-switch configurations, local and global disk drive/printer assignments, manufacturer, model number, serial number, capacity, where and when each item has been purchased, and warranty information. Failures of each device, the type, time, and all the symptoms of the failure should also be reported in this log. From time to time, these symptoms should be compared to find possible correlations between the failures. Also useful on the log is the description of locally maintained or developed software as well as unique features of each software item on the network.

IMPROVING LAN PERFORMANCE

Reducing the load
Keep personal files in the local storage. Get the heavy users, such as CAD stations, off the network if necessary.

Caching
Network performance is affected more by the number of times data is moved than by the amount of data moved. Therefore, network performance can be substantially improved by disk caching.

Filing Structures
In order to speed up access time to the files, keep directories as flat as possible.

Hard-Disk Storage
Hard-disk transfer speeds are rated between 5 to 10 Mbits/sec. In reality, these rates are much lower. By interleaving the sectors, speed may be improved. But, in a multiuser environment, using several smaller hard disks, rather than one large one, may dramatically improve the access speed.

Drive Technology
Drive technology offers a conventional, slower stepping-motor method, or a faster, more expensive voice-coil head-positioning method. The latter method seeks the position two or three times faster than the first method.

Faster CPU's
Faster disk drives are only part of the solution. Also needed is a faster processor that can keep up with faster data rates. Obviously, overall network speed will improve by changing the clock of the processor as well as its type.

Cable
Plan the network applications carefully before laying out the cable. Much higher performances may be obtained by dividing the network into clusters of subsystems and providing gateway devices in between. By its very nature, the TurboDOS Constellation Network allows the network planners to consolidate one application group within a multiuser cluster, thus reducing the load on the network.

Printers and other Peripherals
Unlike hard disks, where speed and performance may be improved by using several smaller units, the issue with printers is quite different. Larger, heavy-duty, high-speed printers are generally more efficient and also handle forms much more reliably. However, the smaller needs of local stations may be better satisfied with inexpensive, lightweight printers.

SECURITY

Security for a microcomputer local area network is a relatively new field. All features of the electronic media which are desirable to a user also make it vulnerable to theft and damage. A stand-alone computer is easy to secure. With its diskettes locked away, it cannot be tampered with. In local area networks, however, the ease of use is also the cause of its misuse. Before any steps can be taken in promoting security of the network, a proper risk analysis should be carried out to determine risks involved and the cost of correcting potential problems. For LANs, these strategies should be considered: physical security, personal identification, encryption, the diskless station, cable radiation, and call-back.

INSTALLATION

A network installation is inherently more complex than a stand-alone computer. Unlike a single personal computer, a network is not removable. Most of the obscure bugs which may plague a network can be prevented by following proper installation procedures. A few guidelines for network installation follow:

Consider the network plan, the building and electrical codes, all elements of the network, the environmental conditions, the building design, and any future expansion plans.

Keep an installation log indicating the types of parts and connectors used and their supplier. Prepare a schematic diagram indicating locations of hubs, connectors, taps, and repeaters. This log and schematic diagram should be updated showing any change on the network.

Consider future growth, modifications, and periodic repairs before laying out the cable. Plan carefully, with ease of access in mind, the routing of the cable. The type of cable to be used is generally dictated by the manufacturer. In high noise areas, triax or twinax cables may be substituted with coax cables. When kinked, coax cables offer different impedance and capacitance. The difference between kinking and bending is a matter of degree. As a rule of thumb, a minimum bend radius is 5 times the cable diameter.

Test the cables before and after installation using at least a simple ohmmeter test. In order to reduce interference, the routing of the LAN cable should be placed away from power lines. The danger of high voltage may be prevented by following proper grounding methods.

Increased number of disk/file servers, hard disks, or printers, as well as duplicate routing of the cable, will result in higher fault tolerance of the system. However, weigh the cost of the installation against the benefits that will be gained.

PERFORMANCE

Performance of the network can be improved by the measures listed on the facing page.

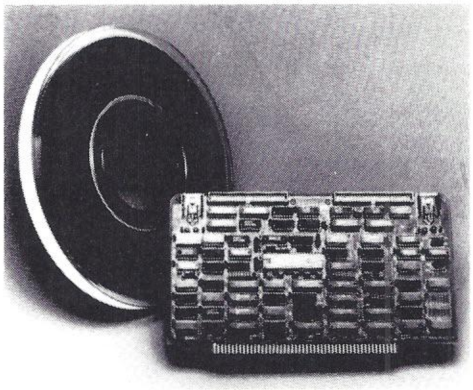
INTERNETTING

Internetting is the process of connecting networks. It permits data to move freely among a large number of networks and populations. A *bridge* is a device which links local networks using identical protocols. Dissimilar networks are connected through *gateways*.

Basically, a **bridge** retransmits every packet of information from one

S-100 USERS

INTERFACE 9-TRACK TAPE DRIVES



With the DTI - DMA Tape-Unit Interface

- **Transfers data via DMA up to 200K bytes per second**
- **Allows full control over all tape drive functions**
- **Meets IEEE S-100 standard**
- **Software available**

For further information contact:

SYSTEM PLANNING CORPORATION
1425 N. Quincy Street, Arlington, VA 22207
(703) 276-4713

Learn to program in DATAFLEX

The most powerful single- and multi-user database development software

Attend our intensive one and two-day seminars

Call for details and schedules

(415) 495-5999

COCHLIN COMPUTER SYSTEMS
330 TOWNSEND STREET #107 SAN FRANCISCO, CA 94107

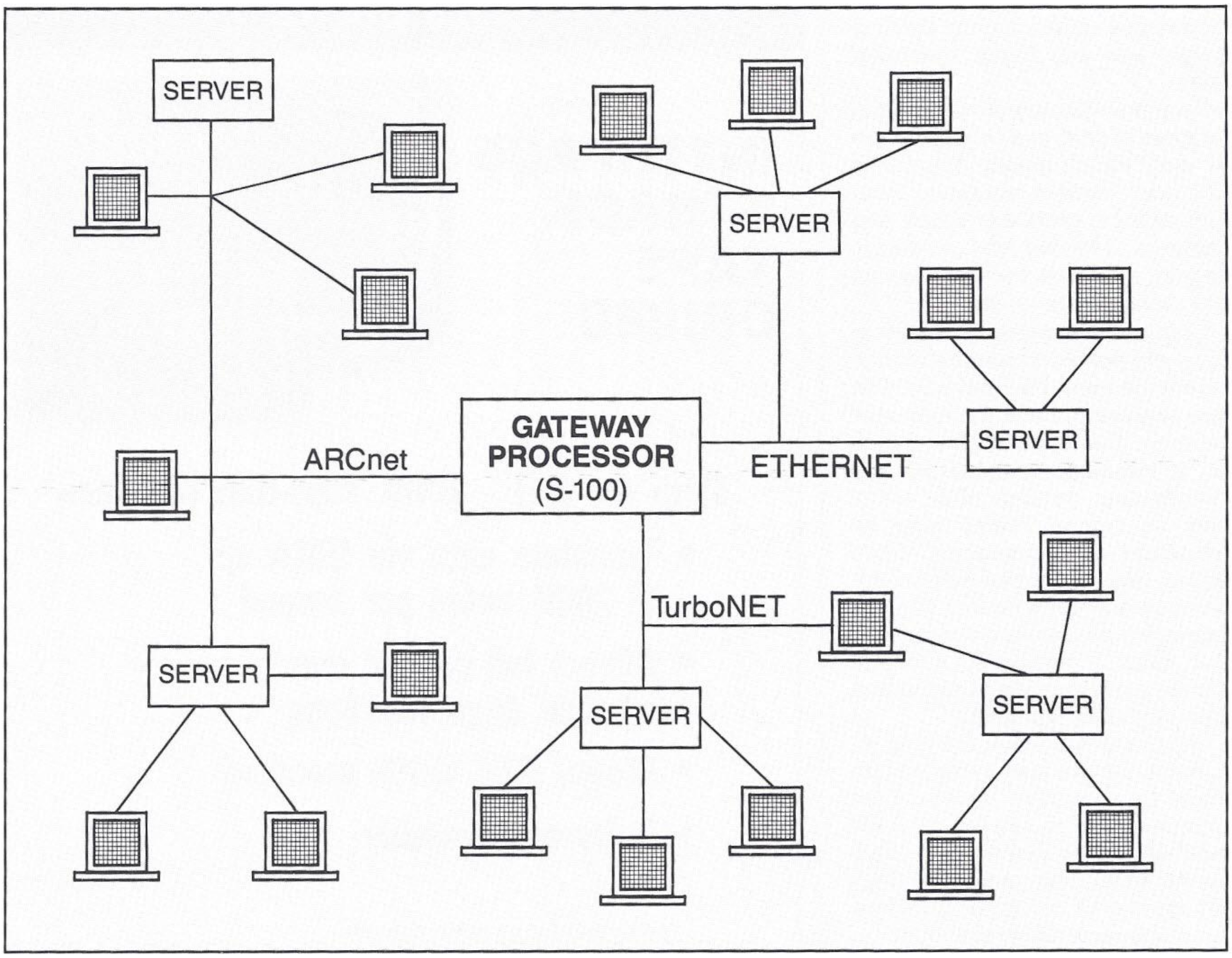


Figure 5. *Different LANs can be linked together through a Gateway Processor.*

network to another. By using bridges, a giant LAN may be divided into more manageable smaller LANs that will provide better overall performance and security. Bridges also increase the distance a network can cover.

A **gateway** receives data transmission from one network and recreates it in the format used by the network of the addressed node point. This means restructuring the message to conform to the size differences and adjusting it to different access and routing schemes. Possible differences in network speeds require the buffering capabilities of gateways. The most difficult service performed by the gateway is addressing, as each device addressed must be stored. Then, this information must be interpreted and properly readdressed.

Due to the complex tasks performed, most bridges and gateways are designed with coprocessors on board.

In 1976, the CCITT (Consultative Committee for International Telegraph and Telephone) adopted Recommendation X.25 which has become the most widely used interface of public networks. In the U.S., three different types of X.25 networks are available to the public: Tymnet, Telenet, and AT&T. Only very large companies use other systems for wide area networking, usually with a private X.25 network or with IBM's System Network Architecture (SNA).

Access to public networks can be done either through gateways that provide X.25 protocol or through strictly-asynchronous modems. The expected traffic load and the cost of the method determines the final choice of access. Another alternative for public networking is leasing a dedicated line from a phone company. But this alternative is much more expensive.

Commercial gateway products offer stand-alone gateway processors or cards which fit into disk/file servers. Depending upon the software used, a Gateway Processor may support either X.25 or SNA protocol, or both.

To access remote locations from the network, a few user stations can be provided with 300/1200/2400-baud modems. Once the data from remote locations are pulled down by these stations, these data may become available for the rest of the users on the network. With the reduction in prices of 9600-baud modems, not only will waiting time be reduced but also the cost of connecting the LAN to mainframe data bases. ■

S-100 Journal thanks Teletek for assistance in the preparation of this article. For more information on the products mentioned, send an Editorial Feature Reply Card to Teletek, 4600 Pell Drive, Sacramento, CA 95838.

SUBSCRIBE

S-100 journal
THE EXCLUSIVE
MAGAZINE
ON S-100 SYSTEMS

PLEASE ENTER MY SUBSCRIPTION TO S-100 JOURNAL.

☐ $14.00 for 1 year (4 issues) ☐ $25.00 for 2 years (8 issues)

(Canada and Mexico, add $3.00 per year. Other countries add $6.00 per year)

☐Bill me (USA only) ☐My payment is enclosed (1 extra issue free)

NAME: ____

ADDRESS: ______________________________________

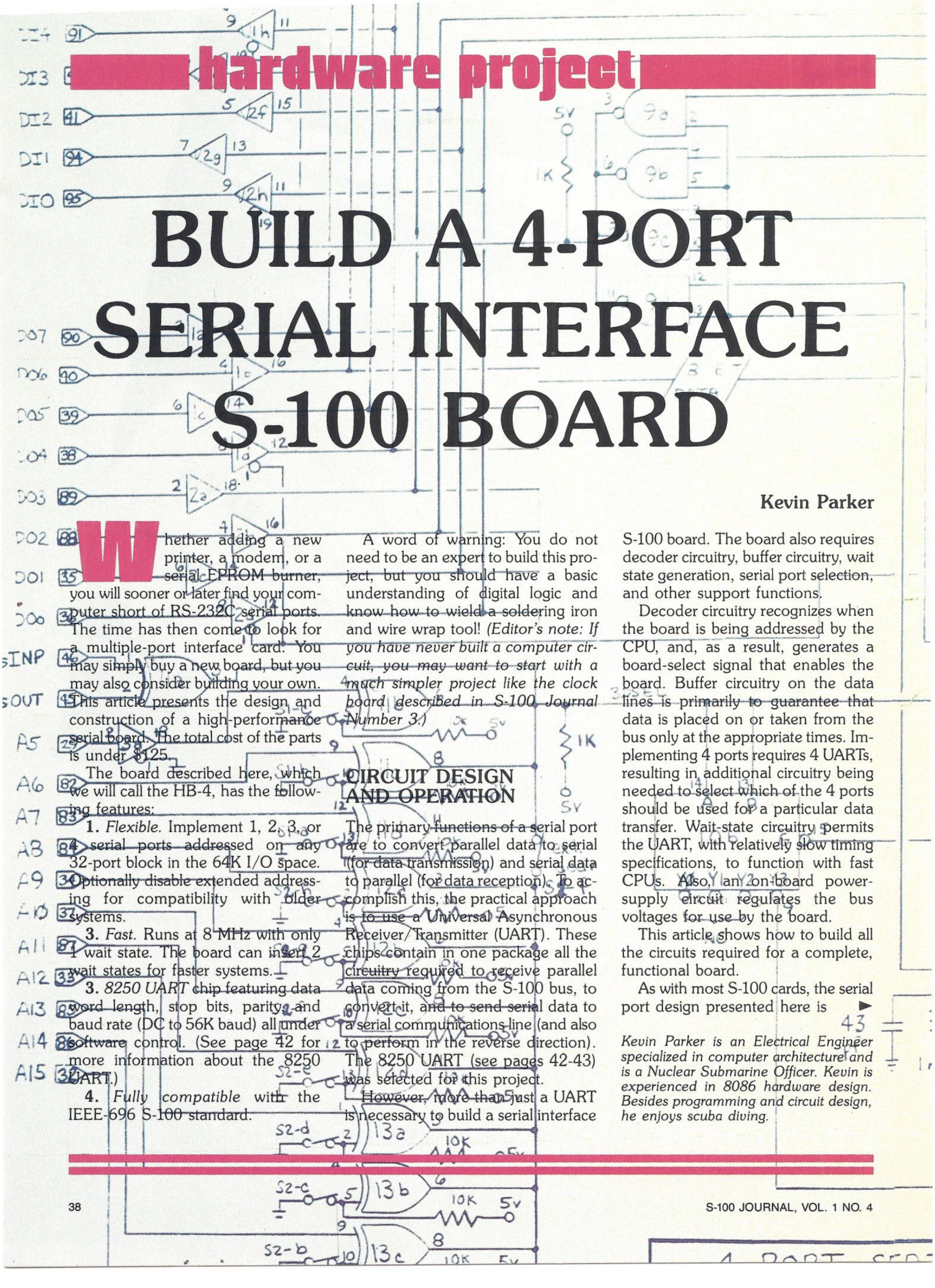
____

PLEASE MAIL TO: S-100 Journal, PO Box 1914, Orem, UT 84057

FALL 86

hardware project

BUILD A 4-PORT SERIAL INTERFACE S-100 BOARD



Kevin Parker

Whether adding a new printer, a modem, or a serial EPROM burner, you will sooner or later find your computer short of RS-232C serial ports. The time has then come to look for a multiple-port interface card. You may simply buy a new board, but you may also consider building your own. This article presents the design and construction of a high-performance serial board. The total cost of the parts is under $125.

The board described here, which we will call the HB-4, has the following features:

1. *Flexible.* Implement 1, 2, 3, or 4 serial ports addressed on any 32-port block in the 64K I/O space. Optionally disable extended addressing for compatibility with older systems.

3. *Fast.* Runs at 8 MHz with only 1 wait state. The board can insert 2 wait states for faster systems.

3. *8250 UART* chip featuring data word length, stop bits, parity, and baud rate (DC to 56K baud) all under software control. (See page 42 for more information about the 8250 UART.)

4. *Fully compatible* with the IEEE-696 S-100 standard.

A word of warning: You do not need to be an expert to build this project, but you should have a basic understanding of digital logic and know how to wield a soldering iron and wire wrap tool! *(Editor's note: If you have never built a computer circuit, you may want to start with a much simpler project like the clock board described in S-100 Journal Number 3.)*

CIRCUIT DESIGN AND OPERATION

The primary functions of a serial port are to convert parallel data to serial (for data transmission) and serial data to parallel (for data reception). To accomplish this, the practical approach is to use a Universal Asynchronous Receiver/Transmitter (UART). These chips contain in one package all the circuitry required to receive parallel data coming from the S-100 bus, to convert it, and to send serial data to a serial communications line (and also to perform in the reverse direction). The 8250 UART (see pages 42-43) was selected for this project.

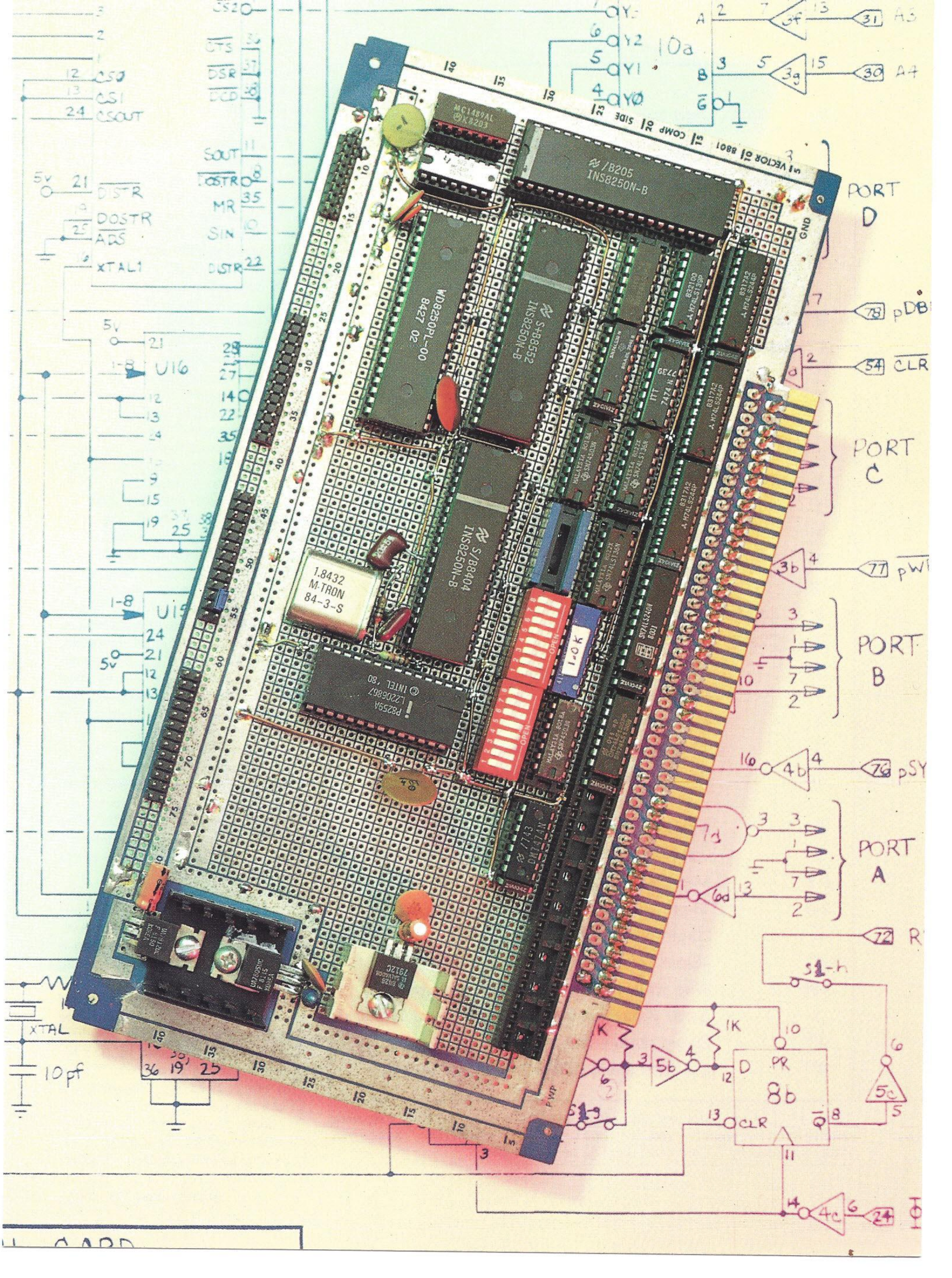
However, more than just a UART is necessary to build a serial interface S-100 board. The board also requires decoder circuitry, buffer circuitry, wait state generation, serial port selection, and other support functions.

Decoder circuitry recognizes when the board is being addressed by the CPU, and, as a result, generates a board-select signal that enables the board. Buffer circuitry on the data lines is primarily to guarantee that data is placed on or taken from the bus only at the appropriate times. Implementing 4 ports requires 4 UARTs, resulting in additional circuitry being needed to select which of the 4 ports should be used for a particular data transfer. Wait-state circuitry permits the UART, with relatively slow timing specifications, to function with fast CPUs. Also, an on-board power-supply circuit regulates the bus voltages for use by the board.

This article shows how to build all the circuits required for a complete, functional board.

As with most S-100 cards, the serial port design presented here is ►

Kevin Parker is an Electrical Engineer specialized in computer architecture and is a Nuclear Submarine Officer. Kevin is experienced in 8086 hardware design. Besides programming and circuit design, he enjoys scuba diving.


CS0
CS1
CSOUT
DISTR
DOSTR
ADS
XTAL1
CTS
DSR
DCD
SOUT
DOSTR
MR
SIN
DISTR
U16
U15
5v
XTAL
10pf
Y2
Y1
Y0
10a
PORT D
pDBI
CLR
PORT C
pWR
PORT B
pSYNC
PORT A
7d
6d
5b
8b
5c
4c
1K
D
PR
CLR
Q
S1-h
S1-g
INS8250N-B
WD8250PL-00
S/B8552 INS8250N-B
S/B8404 INS8250N-B
MC1489AL
P8259A
INTEL '80
1.8432 M-TRON 84-3-S
OPEN
1.0K
79120
DM7474N
GND
PWR
COMP SIDE
VECTOR 8801
CARD

Qty.	ID	Part Description
3	U1-U3	74LS244 Non-inverting octal driver
1	U4	74LS240 Inverting octal driver
1	U5	7406 O.C. Hex inverter
1	U8	74ALS74 Dual D flip flop
1	U9	74LS03 Quad O.C. 2-input NAND gate
1	U10	74LS139 Dual 2/4 decoder
3*	U11-U13	74LS136 Quad O.C. XOR gate
4**	U14-U17	8250A or 8250B UART chip
1	U7	MC1488 Quad line driver
1	U6	MC1489 Quad line receiver
1		7805 5 volt regulator (TO-220 case)
1		7812 12 volt regulator (TO-220 case)
1		7912 −12 volt regulator (TO-220 case)
1		10K resistor array
1		1K resistor array
12		.1 uf capacitors
3	C1,3,5	10 uf/25v capacitor
3	C2,4,6	.05uf disk capacitor
1		1.8432 MHz crystal
1		1.5K resistor
1		1.0 megohm resistor
1		40-60 pf capacitor (mica or mylar)
1		10-30 pf capacitor (mica or mylar)
2*		8-position dip switch
3		Voltage regulator heat sink (TO-220 case)
6-8		14-pin wire wrap sockets
2-3		16-pin wire wrap sockets
4		20-pin wire wrap sockets
4**		40-pin wire wrap sockets
~80		Wire wrap pins — Vector T49 or equivalent
1		S-100 Breadboard — Vector 8801 or equivalent
4**		26-pin double row header
4**		Interface cable (DB25S to 26-pin socket)

Table 1. *List of parts for the HB-4 serial interface card. If extended addressing is not implemented, only one of the parts indicated by * is required. Parts indicated by ** are one required per serial port.*

not complex. The challenge when designing S-100 bus circuits is to minimize part count while maximizing speed.

A more detailed explanation of the HB-4 circuit design is presented next. Refer to the diagram on pages 46-47 as we discuss each circuit.

The data-input bus buffer (U1e-h, U2e-h) is enabled, through U9, by the S-100 signal pDBIN and the CSOUT signal from each 8250 UART. The DDIS output from each 8250 is not used to enable the buffer; doing so would insert an additional 300 nsec delay (due to 8250B timing specifications) which would require adding 2 wait states to each read cycle — for an 8 MHz system. If you do not install all four UARTs, you must tie the resulting unused U9 inputs to ground in order to ensure that the outputs of the data-input bus buffer do not remain active continuously.

Open-collector gates U11-U13 are connected in a 'wire NOR' configuration to generate the board-select signal (BDSEL) from A5-A15 and from the input/output bus status signals. Notice that shutting each address-bit-select switch (S1a-c, S2a-h) requires the corresponding address bit to be a '1' in order to select the board. Opening switch S1-f will disable extended addressing. You may omit U12, U13, S2, and the corresponding resistors if you do not anticipate ever using extended addressing.

Individual 8250s are selected on the board by using a 1-of-4 decoder (U10a). A0-A2, from the S-100 address bus, are used to access individual registers in each 8250. The ADS* input on each 8250 is tied low since the S-100 address bus is not multiplexed.

Data-output bus buffering is provided by U1a-d and U2a-d. The data bus pins on the 8250 are bidirectional, hence, board data-bus contention is eliminated by enabling the buffer with the S-100 data-write strobe (pWR*).

The wait state circuit is active during any I/O cycle in which the board is selected. The D flip flops (U8a and b) are cleared by pSYNC at the beginning of each bus cycle. This brings RDY low, signaling the CPU that a wait state is to be inserted. Assuming S1-g is open, the next bus clock cycle will set flip flop U8a causing a second wait-state to be inserted.

Flip-flop pin 8 will be reset during the following bus clock cycle, signaling the CPU that no more wait states are required. If S1-g is closed, only 1 wait state is inserted. Wait states will not be inserted if S1-h is opened.

On page 47, each serial port consists of four RS-232 lines. Additional lines can be connected as in page 48.

All bus-timing and strobe signals are Schottky buffered in order to clean up pulse shape and minimize bus loading.

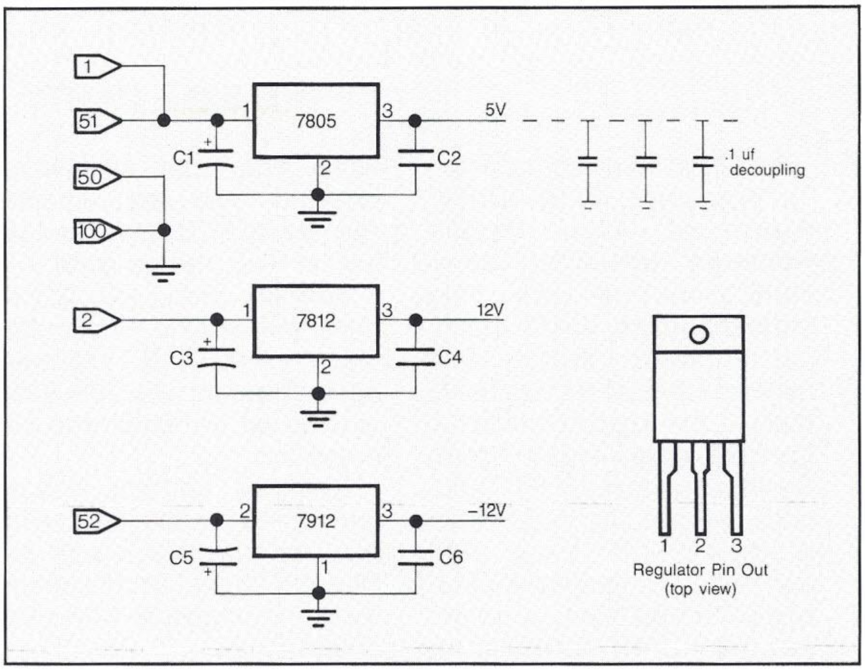


Figure 1. *Schematics of the power supply.*

CONSTRUCTION

First gather all the parts listed in Table 1. Substitute 1/4-watt resistors for the resistor arrays if necessary. Table 2 lists parts that may be deleted for various board configurations.

I constructed the circuit on a Vector 8801 S-100 breadboard card using wire wrap pins and sockets. Start out by installing the voltage regulators on heat sinks in the space provided on the left side of the board. You will need to drill a hole for the third regulator. Mount the filter capacitors C1-C6 close to the regulators to minimize lead length. See Figures 1 and 2. Cut traces and install jumpers as necessary to establish power and ground buses around the edges of the breadboard. Insert and solder wire wrap pins in the holes provided across the bottom of the board for the S-100 bus contacts. It is not necessary to install a pin for every contact; refer to pages 46-47 for the minimum number of pins required.

The layout of parts is not critical, but I found the project easier to construct by locating the bus drivers at the bottom of the board and the RS-232 drivers/receivers at the top (Figure 2). Begin laying out the board by inserting (do not solder yet!) wire wrap sockets for the ICs. I couldn't find any 20-pin wire wrap sockets (couldn't wait for mail order!) for the bus drivers, so I put several Radio Shack 16-pin wire wrap sockets end to end to make a long continuous socket across the bottom of the board. Next, extend the power-supply bus lines from the edge of the board using 18 or 20-gauge solid conductor wire and wire wrap pins. The objective, when laying out this bus work, is to bring the power supply rails near the supply and return pins of each integrated circuit. Jockey the wire wrap sockets and bus work around until you have a suitable layout. Make certain that you leave room between the sockets for .1uf power-supply decoupling capacitors. There should be a capacitor for every 3 or 4 integrated circuits. A good rule of thumb is to disperse the capacitors such that there is a capacitor within 1-2 inches of every IC. The decoupling capacitors connect to 5V and ground.

Once you are satisfied with the board layout, solder the bus work, capacitors, and sockets in place. **Don't** solder **every** pin on each socket to the board. Instead, solder only 2 pins on each socket to make removal easier in the event you ever decide to reuse the sockets.

At this point it is a good idea to plug the board into your computer and measure voltages on the regulators and supply lines. Finding regulator wiring problems now will save countless hours of frustration later.

Finish up soldering by installing the crystal and associated circuitry (tank circuit) on wire wrap pins. Keep the leads as short as possible. If you couldn't find resistor packs, install discrete resistors and wire wrap pins at a convenient location. Solder the 26-pin headers across the top of the board.

Next install the ICs (except for the 8250s) in their sockets, taking care to

Feature Altered	Components Eliminated	Remarks
Implement only 1 serial port	U9, U15, U16, and U17	Connect pin 24 of U14 to pin 19 of U1/U2 via spare inverter.
Implement 3 (2) serial ports	U17 (and U16)	Connect unused inputs of U9 to ground.
No wait state generator	U5 and U8	—
No extended addressing	U12, U13, and S2	—

Table 2. *Board modification summary for implementing fewer features.*

THE 8250 UART

National Semiconductor Corporation's 8250 UART (Universal Asynchronous Receiver/Transmitter) communications chip costs a little more than other UART chips, but its power, flexibility, and minimal interface requirements make it the logical choice for the HB-4 serial interface board. Besides normal UART functions, the chip also implements programmable baud rate generation, modem control, and prioritized interrupt control for up to 4 UART-generated interrupts. All of these features are programmable through control registers (see facing page). These registers are accessed via several port addresses (7 ports for the 8250B, 8 ports for the 8250A). Register functions and software initialization requirements are covered in more detail in the text.

The 8250 chip is available in a few different flavors. Our serial interface board was built with the 8250B, but any other version could have been used. The primary differences between the 8250A and the 8250B are that the A version is faster and contains a scratchpad register.

8250 features

Chip select (pins 12-14) — up to 3 separate inputs can be used to select the chip. Any unused inputs must be tied to their *active* state.

Read and write strobes (pins 18, 19, 21, 22) — separate read and write strobe inputs are provided. Notice the presence of both an active-low and an active-high input, minimizing external logic requirements. Any unused inputs should be tied to their *inactive* state.

Address strobe (pin 25) — this input is used to latch the chip-select and register-address inputs when the 8250 is used on a multiplexed address bus.

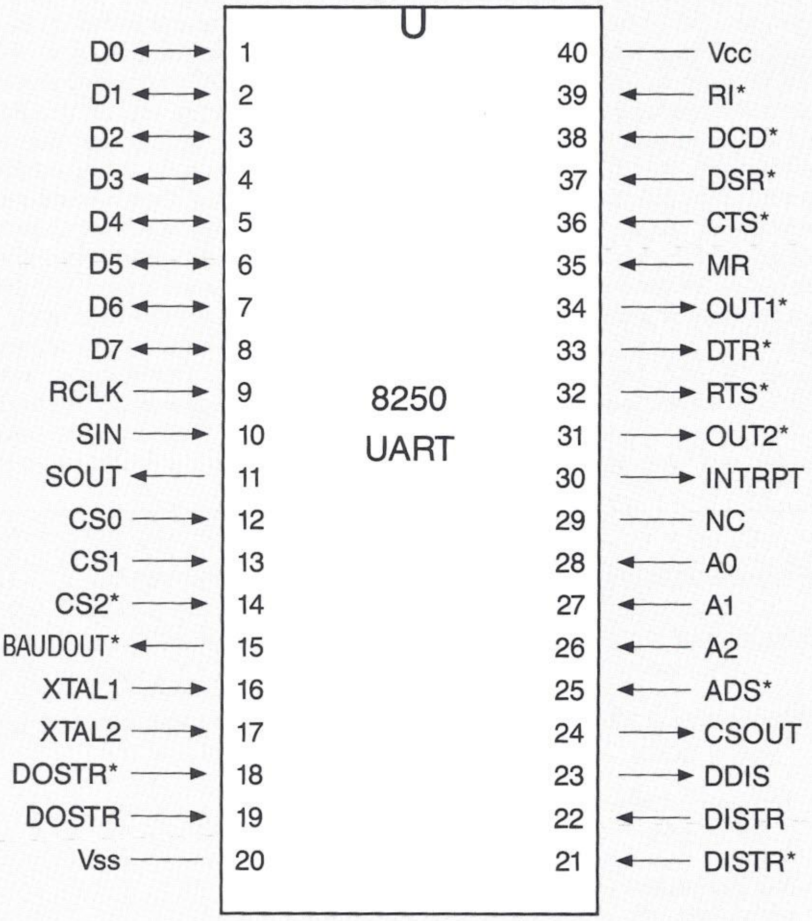
Register address inputs (pins 26-28) — these three inputs, in conjunction with the DLA bit, control access to the 9 internal registers (10 in the 8250A chip).

Modem interface (pins 36-39, 32, 33) — several inputs and outputs are provided to ease interfacing with modems. Refer to the National Semiconductor 8250 Data Sheet for details.

Driver disable output (pin 23) — this output can be used to directly disable the S-100 Data Input bus buffer since this signal is active only when the CPU is reading data from the chip.

Chip select output (pin 24) — this output is active whenever CS0, CS1, and CS2* are active. This signal could be used to enable a data bus buffer or a wait state generator.

General purpose outputs (pins 31, 34) — two programmable outputs are provided to allow additional flexibility.



8250 REGISTER ORGANIZATION

Port Address	DLA Bit	Register Name
0	0	Receiver Buffer (read only)
0	0	Transmitter Holding Register (write only)
0	1	LSB of Divisor Latch (read/write)
1	0	Interrupt Enable Register (read/write)
1	1	MSB of Divisor Latch (read/write)
2	x	Interrupt Identification Register (read only)
3	x	Line Control Register (read/write)
4	x	Modem Control Register (read/write)
5	x	Line Status Register (read/write)
6	x	Modem Status Register (read/write)
7	x	—not used— (Scratch Pad Register in the 8250A)

x=don't care

Interrupt Enable Register Organization

D7	D6	D5	D4	D3	D2	D1	D0 (LSB)
				Modem Status Interrupt	Receiver Error Interrupt	Transmit Buffer Interrupt	Data Available Interrupt
0	0	0	0	0 – disabled 1 – enabled	0 – disabled 1 – enabled	0 – disabled 1 – enabled	0 – disabled 1 – enabled

Interrupt Identification Register Organization

D7	D6	D5	D4	D3	D2	D1	D0 (LSB)
					Interrupt Type**		Poll Bit
0	0	0	0	0	00 – modem status 01 – transmitter empty 10 – received data available 11 – error detected in rcvd data		0 – interrupt pending 1 – no interrupts are pending

*** 'Received Data Error' interrupt is highest priority*

Line Control Register Organization

D7	D6	D5	D4	D3	D2	D1	D0 (LSB)
Divisor Latch Access Bit	Break Bit	Stick Parity	Parity Type	Parity Enable	Stop Bits	Data Length	
	0 – releases SOUT 1 – forces SOUT low	0 – disable 1 – enable	0 – odd 1 – even	0 – disable 1 – enable	0 – 1 bit 1 – 2 bits*	00 – 5 bits 01 – 6 bits 10 – 7 bits 11 – 8 bits	

** 1½ bits if data length is 5 bits*

Line Status Register Organization

D7	D6	D5	D4	D3	D2	D1	D0 (LSB)
	Xmitter Shift Reg Status	Xmitter Hold Reg Status	Break Interrupt	Framing Error	Parity Error	Overrun Error	Data Ready
0	0 – busy 1 – empty	0 – full 1 – empty	0 – no break 1 – SIN is held low	0 – none 1 – invalid stop bit detected	0 – none 1 – parity error detected	0 – none 1 – data rcvd befor. previous data read	0 – rcv buffer is empty 1 – data has been received

IC Number	Type	Gnd	5V	12V	-12V
U1, 2, 3	74LS244	10	20	—	—
U4	74LS240	10	20	—	—
U5	7406	7	14	—	—
U6	MC1489	7	14	—	—
U7	MC1488	7	—	14	1
U8	74ALS74	7	14	—	—
U9	74LS03	7	14	—	—
U10	74LS139	8	16	—	—
U11, 12, 13	74LS136	7	14	—	—
U14, 15, 16, 17	8250A/B	20	40	—	—

Table 3. *IC power and ground connections (pin numbers).*

orient them correctly. Now start wire wrapping the board. Work on one IC at a time, double-checking each connection as you go. Refer to Table 3 for power and ground connections to the integrated circuits. Since the 8250 UARTs can draw as much as 80ma, it is a good idea to use two strands of wire for the power and return pins on these ICs. Refer to Table 2 for modification information if you do not intend to implement all 4 serial ports.

When you get to the RS-232 interface at the top of the board, solder the ends of the wires from U6 and U7 to the appropriate pins of the 26-pin headers.

After you are finished wiring the board, carefully inspect both sides for small bits of wire and solder bridges. Next, you will need to obtain four cables (less if you are implementing fewer ports) to connect the RS-232 interface to the computer back panel. If you decide to make the cables yourself, you will need 25- or 26-conductor ribbon cable, four 26-pin IDC socket connectors, and four DB-25 female IDC connectors. Use a bench vise to press the connectors onto the ribbon cable. You should buy the cables if you don't have access to a vise. Using pliers to make cables may ruin the connectors and result in poor electrical connections.

Finally, install the 8250s in their sockets. Be sure to follow standard MOS (semiconductor) handling precautions to prevent any physical or electrostatic damage to the chips. Use a wrist grounding strap and ground mat if available.

INSTALLATION AND TESTING

Before installation, configure the board for your computer by setting the option switches. If your system does not support extended I/O addressing (16-bit I/O port addresses), turn switch S1-f off. Use S1-a, b, c and S2 to set the base address of the board. Notice that shutting a switch requires the corresponding address bit to be a '1.' The S2 switches will have no effect if extended addressing is disabled.

Select the number of wait states required by your system with switches S1-g and S1-h. One wait state is required when the 8250B chip is used at 8-MHz. The HB-4 board can run up to 6 MHz with no wait states if the 8250A chip is used. If you have any doubts, set 2 wait states (S1-g open and S1-h closed) until you know that the board operates correctly. Table 4 summarizes all switch setting options.

Install the complete board and cables in your computer. Power up and boot the system.

The simplest way to test the board is to use DEBUG, SYMDEB, or some other debugger program that allows direct access to I/O ports. Assuming that the board's base address is 0000H, test the board by reading a byte from the Line Status Register of serial port A (port address 5). You should get 60h if the board is operating correctly. If you don't, try reading a byte from port addresses 0-7. If you get FFh from each of these ports, there is a problem in the board-select circuitry, data-input bus buffer, or the 8250 chip itself. Read the Line Status registers of the remaining

Switch	Function	ON	OFF
S1-a	Board address bit 7	A7=1	A7=0
S1-b	Board address bit 6	A6=1	A6=0
S1-c	Board address bit 5	A5=1	A5=0
S1-d,e	—not used—	—	—
S1-f	Extended addressing	Enable	Disable
S1-g	Number of wait states	1 state	2 states
S1-h	Wait state generator	Enable	Disable
S2-a	Board address bit 15	A15=1	A15=0
S2-b	Board address bit 14	A14=1	A14=0
S2-c	Board address bit 13	A13=1	A13=0
S2-d	Board address bit 12	A12=1	A12=0
S2-e	Board address bit 11	A11=1	A11=0
S2-f	Board address bit 10	A10=1	A10=0
S2-g	Board address bit 9	A9=1	A9=0
S2-h	Board address bit 8	A8=1	A8=0

Table 4. *DIP switch settings and functions.*

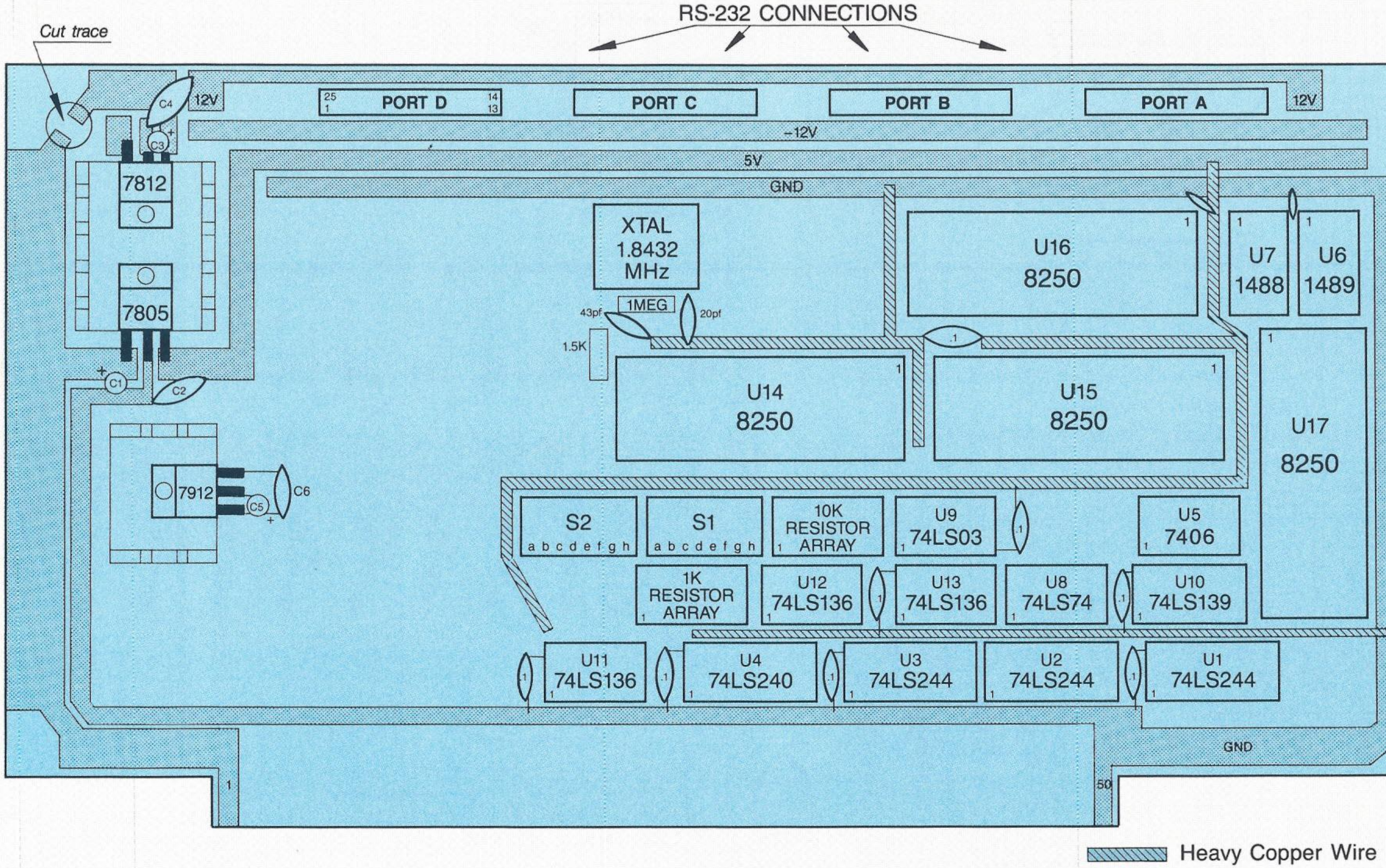
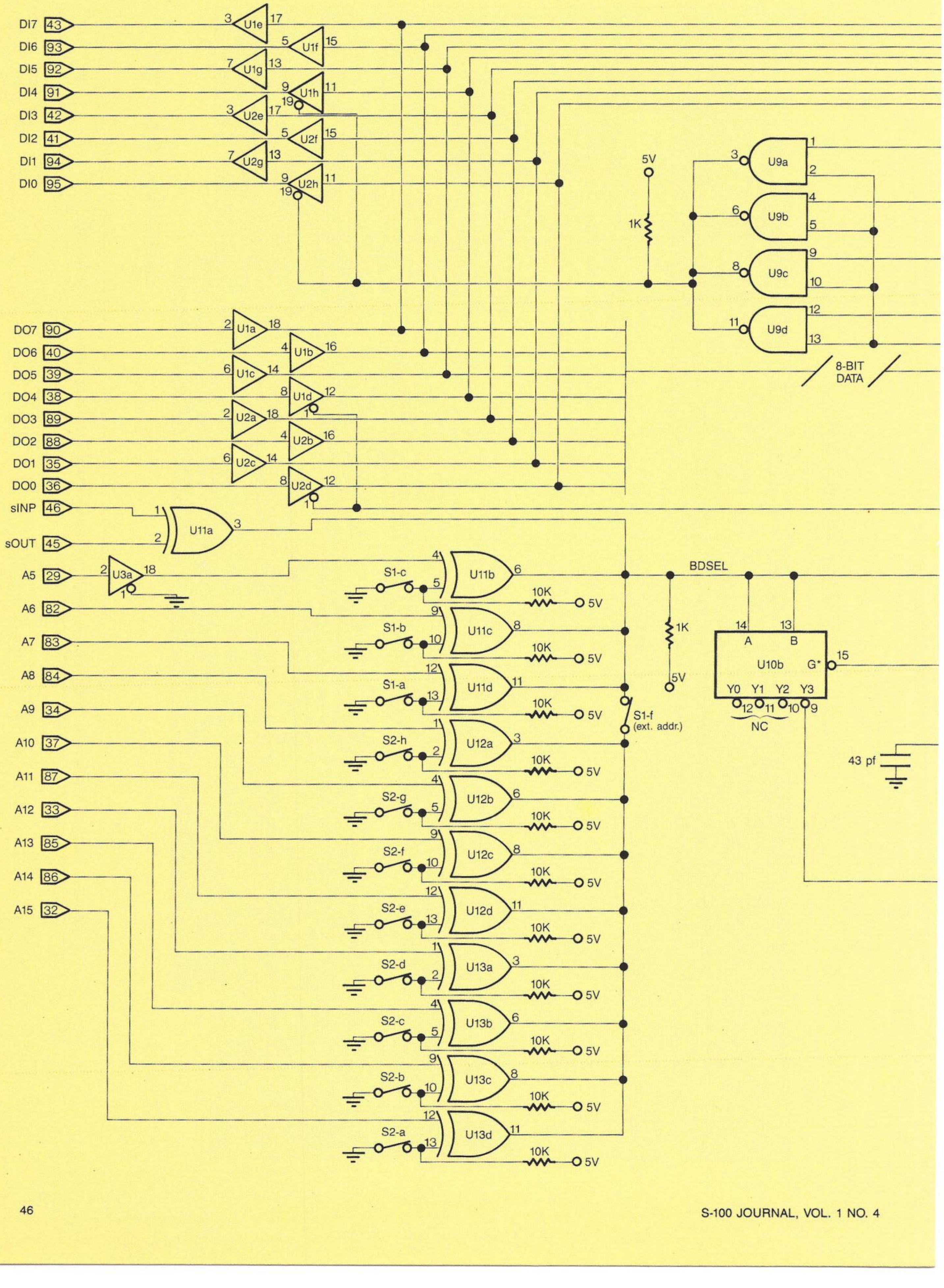
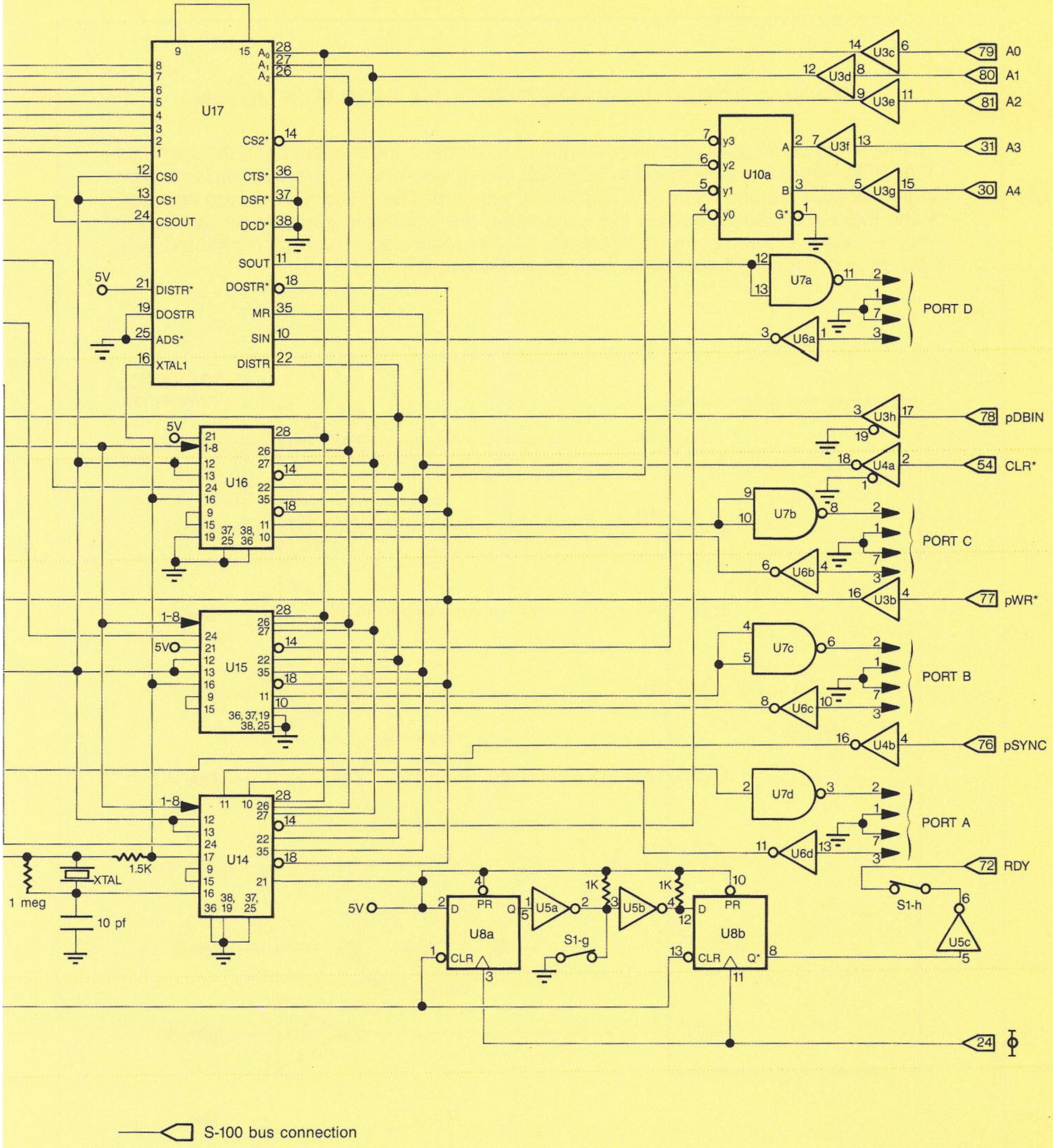

RS-232 CONNECTIONS
Cut trace
PORT D
25
1
14
13
PORT C
PORT B
PORT A
12V
12V
-12V
5V
GND
C4
C3
+
7812
7805
C1
+
C2
7912
C5
+
C6
XTAL
1.8432
MHz
1MEG
43pf
20pf
1.5K
U16
8250
U7
1488
U6
1489
U14
8250
U15
8250
U17
8250
.1
S2
a b c d e f g h
S1
a b c d e f g h
10K
RESISTOR
ARRAY
U9
74LS03
U5
7406
1K
RESISTOR
ARRAY
U12
74LS136
U13
74LS136
U8
74LS74
U10
74LS139
U11
74LS136
U4
74LS240
U3
74LS244
U2
74LS244
U1
74LS244
GND
1
50
▨ Heavy Copper Wire

Figure 2. *Component layout of the HB-4 Serial Interface Board (interrupt circuit not included).*



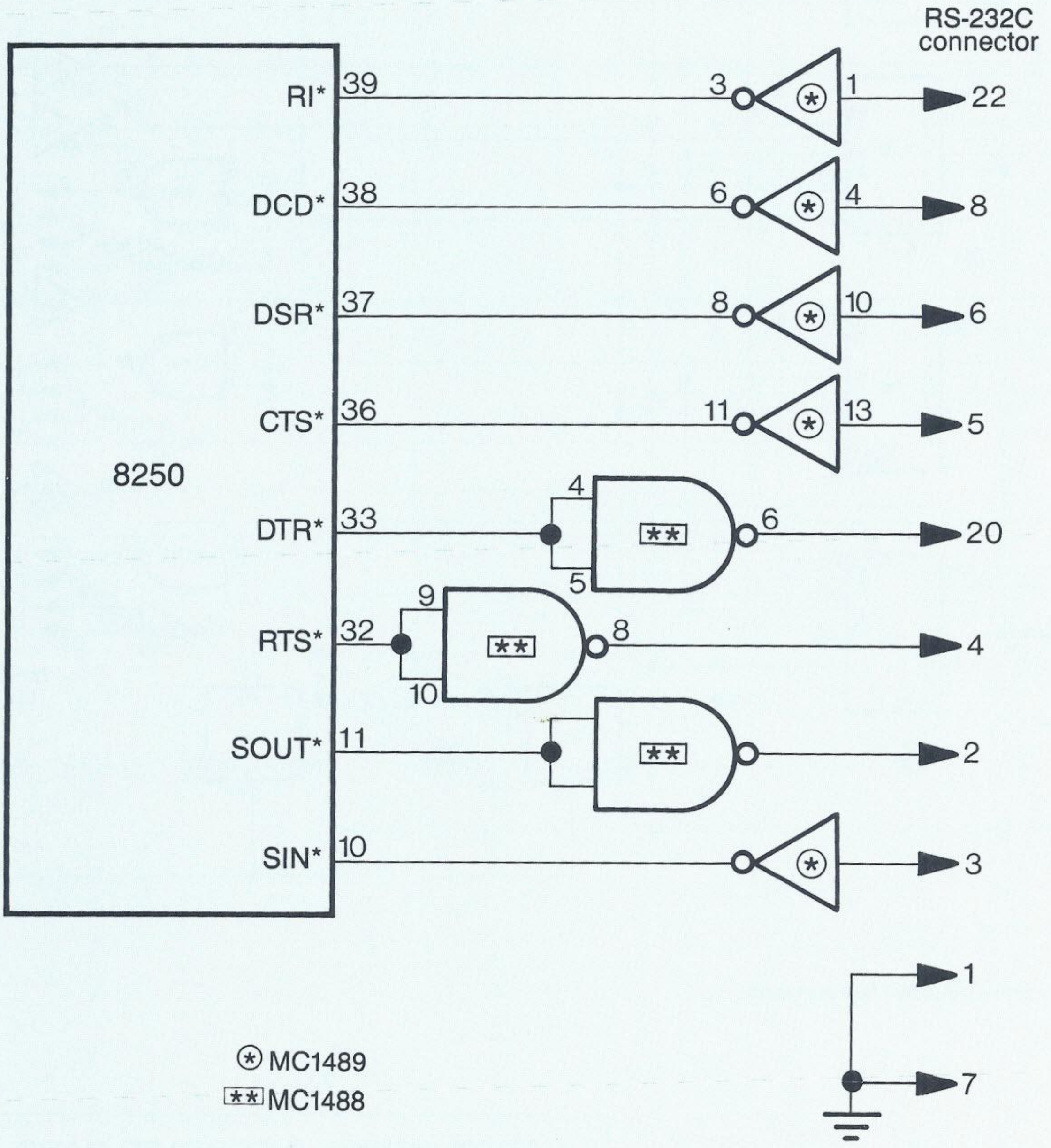
DI7 43
DI6 93
DI5 92
DI4 91
DI3 42
DI2 41
DI1 94
DI0 95
U1e
U1f
U1g
U1h
U2e
U2f
U2g
U2h
5V
1K
U9a
U9b
U9c
U9d
8-BIT DATA
DO7 90
DO6 40
DO5 39
DO4 38
DO3 89
DO2 88
DO1 35
DO0 36
sINP 46
sOUT 45
U1a
U1b
U1c
U1d
U2a
U2b
U2c
U2d
U11a
A5 29
A6 82
A7 83
A8 84
A9 34
A10 37
A11 87
A12 33
A13 85
A14 86
A15 32
U3a
S1-c
S1-b
S1-a
S2-h
S2-g
S2-f
S2-e
S2-d
S2-c
S2-b
S2-a
U11b
U11c
U11d
U12a
U12b
U12c
U12d
U13a
U13b
U13c
U13d
10K
5V
S1-f (ext. addr.)
BDSEL
1K
5V
U10b
A
B
G*
Y0 Y1 Y2 Y3
NC
43 pf


U17
A0
A1
A2
CS2*
CS0
CS1
CSOUT
CTS*
DSR*
DCD*
SOUT
DISTR*
DOSTR*
DOSTR
MR
ADS*
SIN
XTAL1
DISTR
5V
U16
U15
U14
1-8
1.5K
XTAL
1 meg
10 pf
U10a
y3
y2
y1
y0
A
B
G*
U3c
U3d
U3e
U3f
U3g
U3h
U3b
U4a
U4b
U5a
U5b
U5c
U6a
U6b
U6c
U6d
U7a
U7b
U7c
U7d
U8a
U8b
D
PR
Q
CLR
Q*
1K
S1-g
S1-h
79 A0
80 A1
81 A2
31 A3
30 A4
PORT D
78 pDBIN
54 CLR*
PORT C
77 pWR*
PORT B
76 pSYNC
PORT A
72 RDY
24 Φ
S-100 bus connection

4-PORT SERIAL INTERFACE CARD
Designed by Kevin Parker

OPTIONAL CIRCUIT TO CONNECT THE HB-4 CARD TO A MODEM

If any of the serial ports is to be connected to a modem, the modem handshaking lines of the 8250 should be connected to RS-232C line drivers and receivers as shown below. Additional MC1488 and MC1489 chips will be required. The modem inputs are not used by the 8250 to enable or disable the transmitter; the inputs are available for use by the driver software and can generate interrupts if the modem status interrupt is enabled. See the National Semiconductor 8250 data sheet for details.


RS-232C connector
8250
RI* 39
DCD* 38
DSR* 37
CTS* 36
DTR* 33
RTS* 32
SOUT* 11
SIN* 10
3
1
22
6
4
8
8
10
6
11
13
5
4
5
6
20
9
10
8
4
2
3
1
7
⊛ MC1489
** MC1488

8250s using the I/O port address map in Table 5.

Now initialize serial port A by writing, in order, the sequence of bytes listed in Table 6. When you are finished, serial port A will be set up for 9600 baud, 8-bit data, 1 stop bit, and no parity. This will be discussed in more detail later.

Insert a jumper between pins 2 and 3 on the RS-232 connector for serial port A. Read from port 00h to clear the Receiver Buffer. Then perform a loop-back test by writing 35h to port 0. Reading from port 5 should then give you 61h. Next, read from port 0 to get the character you just transmitted (35h). Now read from port 5 and you should get 60H. If you are using DEBUG or SYMDEB, the initialization sequence and loopback test should give results similar to those in Table 7. If the above tests fail, you either have a bad 8250, MC1488, MC1489, or a wiring problem in the associated circuitry.

Port Address	Data Byte
03h	80h
00h	0Ch
01h	00h
03h	03h

Table 6. *Test sequence data to initialize serial port A.*

Port Address Offset	Port Assignment
00h	Port A — Data Register / Divisor Latch (LSB)
01h	Port A — Int. Enable Reg. / Div. Latch (MSB)
02h	Port A — Interrupt Identification Register
03h	Port A — Line Control Register
04h	Port A — Modem Control Register
05h	Port A — Line Status Register
06h	Port A — Modem Status Register
07h	8259 Initialization Command 1 Register*
08h	Port B — Data Register / Divisor Latch (LSB)
09h	Port B — Int. Enable Reg. / Div. Latch (MSB)
0Ah	Port B — Interrupt Identification Register
0Bh	Port B — Line Control Register
0Ch	Port B — Modem Control Register
0Dh	Port B — Line Status Register
0Eh	Port B — Modem Status Register
0Fh	8259 Operational Command 1 Register*
10h	Port C — Data Register / Divisor Latch (LSB)
11h	Port C — Int. Enable Reg. / Div. Latch (LSB)
12h	Port C — Interrupt Identification Register
13h	Port C — Line Control Register
14h	Port C — Modem Control Register
15h	Port C — Line Status Register
16h	Port C — Modem Status Register
17h	not used (Port C Scratch Reg. if 8250A used)
18h	Port D — Data Register / Divisor Latch (LSB)
19h	Port D — Int. Enable Reg. / Div. Latch (MSB)
1Ah	Port D — Interrupt Identification Register
1Bh	Port D — Line Control Register
1Ch	Port D — Modem Control Register
1Dh	Port D — Line Status Register
1Eh	Port D — Modem Status Register
1Fh	not used (Port D Scratch Reg. if 8250A used)

Table 5. *I/O port map for the HB-4 serial board. Ports marked * are not used if an 8259 is not installed (see last section of the article for using the 8259).*

SOFTWARE

Board Initialization

To the programmer, each 8250 UART appears as 7 consecutively addressed I/O ports (8 for the 8250A). For the following discussion, you may want to refer to page 43 where the organization of the registers is summarized. Only the registers that are used in this application will be discussed. The port numbers mentioned in this discussion all refer to the 3-bit addresses (A0-A2) used by each 8250.

The bidirectional data port for the UART is located at address 0 of the 8250. All transmitted and received data for the UART passes through this port. Notice that this port also serves as the *lower half* of the baud-rate Divisor Latch when the DLA bit (D7) in the Line Control Register is set to 1.

Port 1 is the bidirectional Interrupt Enable Register which also serves as the *upper half* of the baud-rate Divisor Latch when the DLA bit (D7) in the Line Control Register is set to 1.

The serial line characteristics are programmed by setting the appropriate bits in the Line Control Register (port 3). This port is also bidirectional, allowing you to directly read line characteristics that you have previously set. When the MSB (Most Significant Bit) of this register is set to 1, reading and writing ports 0 and 1 of the chip will access the baud-rate Divisor Latch.

The Line Status Register, port 5, is used to determine when the UART is ready to transmit and when it has

```
—O 03 80
—O 00 0C
—O 01 00
—O 03 03
—I 00
FF
—O 00 35
—I 05
61
—I 00
35
—I 05
60
```

Table 7. *Example of a DEBUG test session under MS-DOS.*

received a character. This register also contains bits that indicate when errors have been detected in received data. Since the transmitter is double-buffered, there are two transmitter-status bits — one for the actual Transmitter Shift Register, and one for the Transmitter Holding Register. You should use the holding-register status to determine when the 8250 is ready to transmit.

Before the UARTs can do any useful work, they must be initialized by setting the baud rate, word length, number of stop bits, and parity checking. Loading a 16-bit baud-rate divisor in the Divisor Latch (ports 0 and 1) sets the baud rate. The baud-rate divisor is calculated by dividing the 8250 clock frequency by 16 × baud rate. For example, the baud-rate divisor for 9600 baud is:

$$\text{Divisor} = \frac{1{,}843{,}200}{16 \times 9600} = 12 = 0\text{Ch}$$

Word length, stop bits, and parity are set according to the byte written into the Line Control Resister. See page 43 for details. Notice that the most significant bit (the DLA bit) in the Line Control Register must be set (to 1) in order to write the baud-rate divisor to the Divisor Latch. An example of code (for the 8086 CPU) that sets serial port B for 300 baud, 8-bit data, even parity, and 1 stop bit is shown in Table 8. The NOP instructions are required due to the '2.3-microsec minimum write-cycle time' specification of the 8250B. Since the instructions on lines 2-5 take a total of 20 clock cycles (2.5 microsec on an 8-MHz 8086 CPU) to execute, the write-cycle timing specification is met. More NOP instructions would have to be inserted for faster systems. Notice that, once the UART is initialized, the DLA bit must be set to 0 in order to send and receive data.

```
;Initialization code for Serial Port B
MOV      AL,80h
OUT      0Bh,AL          ; setup to access divisor latch of
                                 ; serial port B.
MOV      AL,01h          ; MS byte of baud rate divisor.
NOP                      ; stall for 2.3 microsec (for 8250B).
NOP
OUT      09h,AL          ; MS byte of divisor latch.
MOV      AL,80h          ; LS byte of baud rate divisor.
NOP
NOP
OUT      08h,AL          ; LS byte of divisor latch.
MOV      AL,1Bh          ; even parity, 8-bit data, 1 stop bit.
NOP
NOP
OUT      0Bh,AL          ; set the line characteristics.
```

Table 8. *Example of initialization code. This code sets Port B for 300 baud, 8-bit data, even parity, and 1 stop bit.*

```
             ; Transmit routine for Serial Port A
CHEK_STAT:   IN      AL,05
             TEST    AL,20h         ; is the transmitter holding
             JZ      CHEK_STAT            ; register empty?
             MOV     AL,DL          ; send the data passed
             OUT     00,AL                ; in DL.
```

Table 9. *Example of code for serial output.*

```
             ; Receive routine for Serial Port A
CHEK_RDY:    IN      AL,05
             TEST    AL,01          ; has data been received?
             JZ      CHEK_RDY
             IN      AL,00          ; get the data.
```

Table 10. *Example of code for serial input.*

Input and Output

Input/Output requests in a computer can be handled in either of two ways — *interrupt driven* or *polling*. The interrupt-driven system optimizes CPU time but requires more complex hardware and software. Software considerations for a polled system are discussed next.

In a polled scheme of operation, output requests are handled by querying the port until it is ready to transmit. For an 8250, this is done by waiting until the Transmitter Buffer is empty before transmitting data. One way to do this is to read the Line Status Register until the transmitter-buffer status bit (D5) goes high, signifying that the UART is ready to accept data for transmission. An example of code that will accomplish this is shown in Table 9. If you read the Line Status Register immediately after execution of the last instruction, you would find that the transmitter-buffer status bit would be low until the data was transferred to the Transmitter Shift register. NOP instructions are not required here since the read-cycle specification of the 8250B is 2.2 microsec and the instruction loop takes 31 CPU clock cycles (3.9 microsec) to execute.

Reading data from an 8250 requires waiting until data is received. A way to accomplish this is to periodically read the Line Status Register until the receiver-buffer status bit (D0) is set. An example of code that performs this function is shown in Table 10. Reading a character from the Receiver Buffer of the UART will reset to 0 the receiver-buffer status, overrun-error, parity-error, and framing-error bits in the Line Status Register.

You should write and install a device driver so that your operating system can control I/O to the ports. If you are running MS-DOS version 2.0 or greater, refer to the MS-DOS Programming Reference Manual for details. A sample MS-DOS device driver that provides I/O control-string capability starts on page 17.

ADDING INTERRUPT CIRCUITRY (optional)

One big disadvantage of polled I/O control is its wastage of CPU time. Refering to Table 9, it is apparent that if this code were incorporated in a loop to output a block of data to a UART, the CPU would spend most of its time (>90%) waiting for the UART to transmit the next piece of data. Certainly it would be better if the UART automatically **interrupted** the CPU every time it was ready to transmit the next piece of data, thereby allowing the system to do other work in between data transmissions. This concept is referred to as **multitasking** and allows a computer to perform tasks such as 'background' printing, real-time data collection, and multiple sensor monitoring, all at essentially the same time.

As shown on page 43, the 8250 can generate an interrupt from up to four different conditions. These are listed below in descending priority:

1. *Received Data Error.* This interrupt occurs whenever a framing error, parity error, overrun error, or break interrupt (SIN held at logic-0 level for more than 1-character transmission time) is detected. This interrupt is reset

BE-82 VIDEO BOARD FOR THE S-100 BUS

FEATURES:
- Bit-mapped characters and graphics, 640 by 330 pixel display
- Full Greek set in EPROM, 7 by 9 character, 8 by 11 window
- Characters placed on any raster line, 30 char-line display
- Load a user program in BE-82 RAM for special commands
- Create graphics on one screen while viewing another
- 660 raster lines available for a single drawing
- Effective baud rate to 10,500 in terminal mode

S-100 BUS HACKERS:
- Full documentation, user can customize to special needs
- Purchase items singly, as a group, or choose additions later
- Bare board plus hardware/software documentation..............$135 ☐
- Programmed EPROMS (three 2716s)..................................30 ☐
- Chip set plus hardware components.................................125 ☐
- Assembly and testing..100 ☐

HARDWARE:
- Amdek 310A Video Monitor and Key Tronic keyboard recommended
- S-100 bus system or ZENITH Z-100 Low Profile Computer
- BE-82 VIDEO BOARD acts as a slave device at all times on bus
- Board operation is independent of host processor clock speed

SCIENTISTS AND ENGINEERS:
- DB-WRITE Technical Word Processor.................................$100 ☐

Write your papers using a display that shows exactly what is printed. Operates under CP/M 80 by Digital Research and requires a 64k system. The BE-82 VIDEO BOARD is required to use DB-WRITE. Full documentation and all files provided. Use CP/M to customize keyboard layout by changing jump table, modify printer driver, etc. Printer driver included for the NEC SPINWRITER and the Tech Math/Times Roman thimble. Example of display screen and printout:

$$f(\lambda) = \int_{-\infty}^{\infty} e^{-\beta(\lambda^2+\xi^2)}\left[\frac{\partial^2\rho(\xi)}{\partial\xi^2} + H_0(\xi)\sin^2(\lambda-\xi)\right]d\xi \qquad \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

Position char's $0\;^{2}\;_{4}\;^{6}\;_{8}\;^{10}\;_{8}\;^{6}\;_{4}\;^{2}\;0$ from line

$\alpha - \beta - \gamma - \delta - \varepsilon - \eta - \lambda - \mu - \xi - \pi - \rho - \sigma - \tau - \omega$

Name ______________________________

Address ____________________________

City _______________________________

State ______________ ZIP ______________

☐ Additional Information

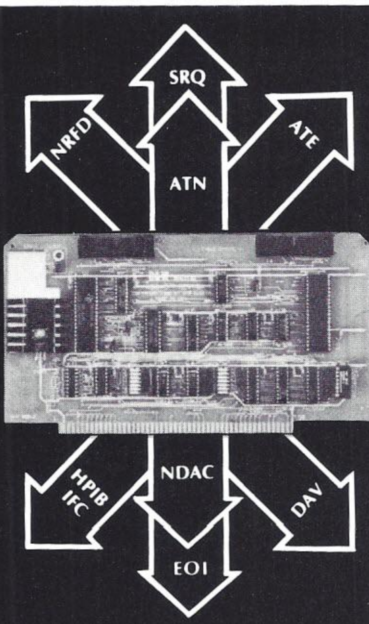
D BAGANOFF SYSTEMS, 3141 DAVID COURT, PALO ALTO, CA 94303

HOMEBREWING COLUMN

Because of the large amount of material present in this issue, we had to leave out the *Homebrewing* section. Rest assured that *Homebrewing* will return.

For those coming in now, *Homebrewing* is a column dedicated to everything that relates to hacking about with S-100 hardware. We publish everything from your frustations and experiences to solid information on how you've put your own system together and how you did (or didn't) overcome problems.

Both articles and letters (amusing ones particularly) are welcome and considered for publication. Send all material to Editor, S-100 Journal, PO Box 1914, Orem, UT 84057.



THE
488+3
IEEE 488 TO S-100
INTERFACE

S-100 ⟷ 488

- Controls IEEE 488 (HP1B) Instruments with an S-100 computer
- Acts as controller or device
- Basic and assembly language drivers supplied
- Meets IEEE 696 specification
- Industrial quality burned in and tested up to 125K bytes/sec under software control 3 parallel ports (8255-5)
- $375

DWD

D&W DIGITAL, INC.
20655 Hathaway Avenue
Hayward, California 94541
(415) 887-5711

by reading the Line Status Register.

2. *Received Data Available.* Occurs whenever a complete data word (5, 6, 7, or 8 bits) is received on the serial input line. Reset this interrupt by reading the Receiver Buffer Register.

3. *Transmitter Empty.* Occurs whenever the Transmitter Holding Register is empty (ready to accept more data for transmission). This interrupt is reset upon writing data to the Transmitter Holding Register or by reading the Interrupt Identification Register (if this was the cause of the interrupt).

4. *Modem Status.* Occurs when any of the CTS*, DSR*, RI*, or DCD* inputs to the 8250 change state. This interrupt is reset upon reading the Modem Status Register.

Since any combination of the above interrupts can occur at the same time, the 8250 contains priority-resolution circuitry so all interrupts get recognized. At any time, the Interrupt Identification Register holds the ID of the highest-priority active interrupt (if enabled). Each of the 4 interrupts can be selectively enabled by setting to 1 the appropriate bit in the Interrupt Enable Register (see page 43).

There are a couple of ways to add interrupt capability to the HB-4. One simple way is to connect the interrupt output from each 8250 to a S-100 bus vectored interrupt line through an open-collector inverter. If all four 8250s were installed, half of the vectored interrupt lines in the system bus would be required. A better way would be to use an 8259A Priority Interrupt Controller. This chip would free 3 bus interrupt lines and, in addition, it would handle another 4 externally-generated interrupts.

The circuitry for adding an interrupt controller is shown in Figure 3. This addition assumes that the bus master (or some other board on the bus) provides the interrupt-cycle bus timing and status signals specified in the IEEE-696 S-100 Bus Standard. Since a detailed explanation of the 8259 is beyond the scope of this article, only highlights of this modification will be covered. You should be familiar with 8259 operation before making this modification.

As shown in Figure 3, A0-A4 are connected such that the 8259's two addressable registers are located at Base+07h and Base+0Fh, thus allowing the board to still occupy the original block of 32 I/O ports. This is of no consequence if 8250B UART chips are used for the serial ports. However, if 8250A chips are used, installation of the interrupt controller will prevent use of the Scratch Pad registers in serial ports A and B.

The A0-A2 lines are also connected to the cascade inputs (CAS0-CAS2) on the 8259 to allow this board to be slaved to the master 8259 on a Seattle Computer Products CPU Support Card. The CAS0-CAS2 pins permit cascading several 8259's to each other for implementing systems with greater than 8 levels of interrupts. If you do not have the SCP board (or some other board with an 8259 master), leave the CAS pins disconnected and program this 8259 in the 'single' mode. See the data sheet or Intel Application Note AP-59 for details.

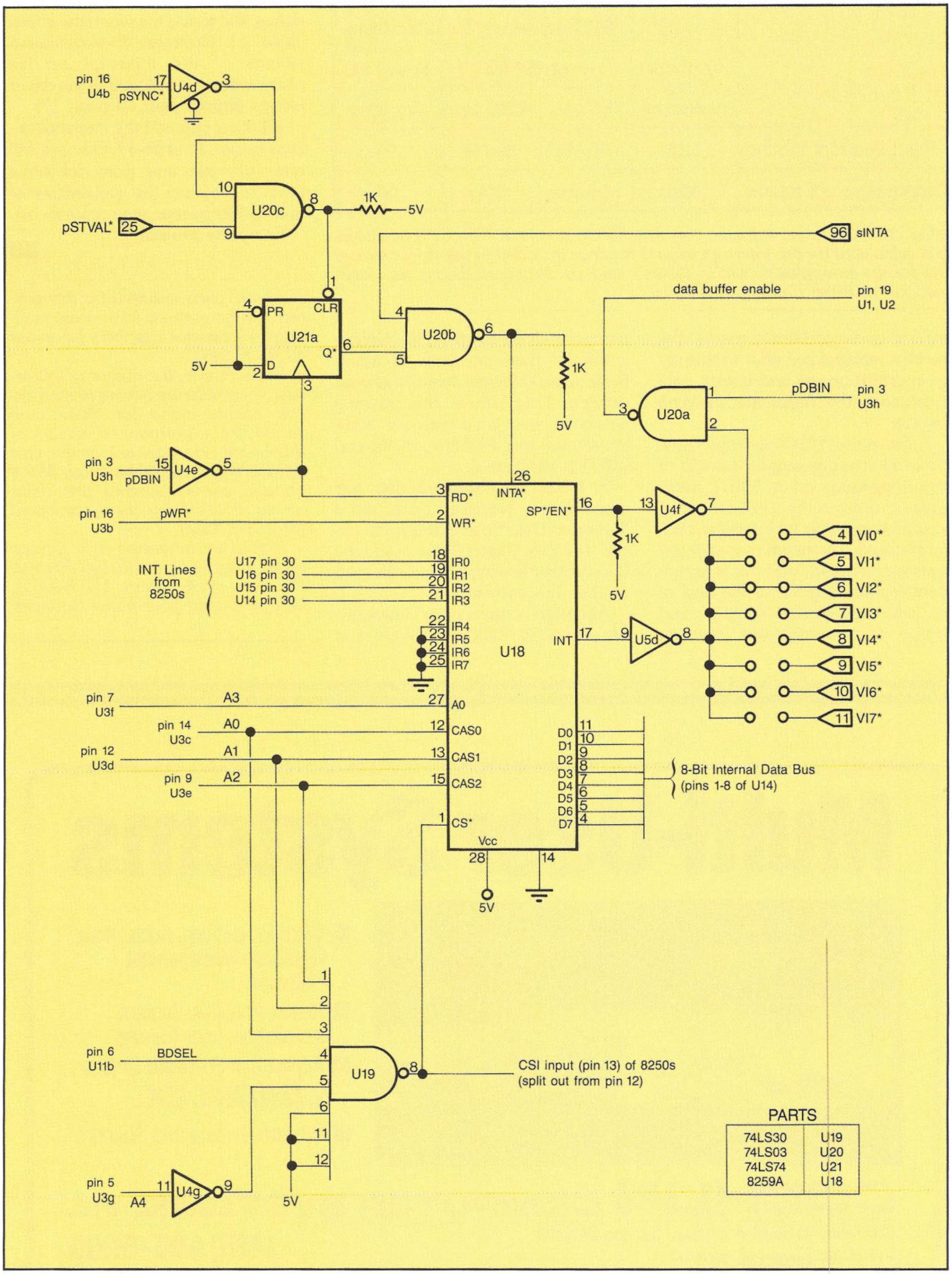
A flip flop (U21a) is used to stretch the INTA* pulse sent to the 8259. The resulting pulse starts at the beginning of the bus cycle (coincident with pSTVAL* and pSYNC) and ends with the trailing edge of pDBIN. This is necessary in order to meet the minimum INTA* pulse-width requirement of the 8259.

The interrupt output (INT) of the 8259 can be connected to any one of the S-100 vectored interrupt lines through an open collector gate as shown. If this 8259 will be slaved to a master 8259 (residing on a different board), select an interrupt line such that the interrupt output of this slave is **not** connected to the IR0 line of the master 8259 (unless all other lines have slaves attached). If no system master 8259 exists, connect the interrupt output of the HB-4's 8259 to the S-100 INT* line (S-100 pin 73).

The SP*/EN* line (pin 16) is used to enable the data-input bus buffer during interrupt-acknowledge bus cycles. This also allows assigning master or slave status to the 8259 during software initialization.

When writing interrupt service routines for the 8250, the following points must be observed (these lessons were learned the hard way!):

1. Manipulating the transmitter-interrupt enable bit (D1) may terminate any data transmission that is in progress. This is true even if you attempt to set the transmitter-interrupt enable bit when it is already set. In


pin 16
U4b
pSYNC*
17
U4d
3
10
U20c
8
1K
5V
pSTVAL*
25
9
96
sINTA
1
4
PR
CLR
U21a
Q*
6
5V
D
2
3
U20b
5
1K
5V
data buffer enable
pin 19
U1, U2
U20a
pDBIN
pin 3
U3h
pin 3
U3h
15
U4e
pDBIN
pin 16
U3b
pWR*
26
INTA*
RD*
WR*
SP*/EN*
16
13
U4f
7
1K
5V
INT Lines
from
8250s
U17 pin 30
U16 pin 30
U15 pin 30
U14 pin 30
18
19
20
21
IR0
IR1
IR2
IR3
22
23
24
25
IR4
IR5
IR6
IR7
U18
INT
17
9
U5d
8
4 VI0*
5 VI1*
6 VI2*
7 VI3*
8 VI4*
9 VI5*
10 VI6*
11 VI7*
pin 7
U3f
A3
27
A0
pin 14
U3c
A0
12
CAS0
pin 12
U3d
A1
13
CAS1
pin 9
U3e
A2
15
CAS2
1
CS*
D0
D1
D2
D3
D4
D5
D6
D7
11
10
9
8
7
6
5
4
8-Bit Internal Data Bus
(pins 1-8 of U14)
Vcc
28
14
5V
pin 6
U11b
BDSEL
U19
CSI input (pin 13) of 8250s
(split out from pin 12)
6
11
12
5V
pin 5
U3g
A4
11
U4g
9

PARTS

74LS30	U19
74LS03	U20
74LS74	U21
8259A	U18

Figure 3. *Circuit modification to add an 8259 Interrupt Controller to the HB-4 serial interface board.*

	Execution Time (microsecs)	CPU availability for other tasks		
		Interrupt I/O		Polled I/O
		1200 baud	19,200 baud	Any baud
Input interrupt routine	53.6	99.4%	89.7%	0%
Output interrupt routine	48.3	99.4%	91%	0%

Table 11. *I/O-driver software performance comparison. The CPU availability is calculated for the interrupt service routines included in the driver listings and for the examples shown in Tables 9 and 10. Numbers shown are worst-case for an 8086 running at 8 MHz.*

order to avoid problems, ensure that this bit is manipulated **after** a transmitter interrupt occurs and **before** data is written to the Transmitter Holding Register.

2. Since the 8250 interrupt signal is a level-sensitive output (instead of edge-triggered), other UART interrupts occuring at the same time may go unnoticed while an interrupt is being serviced. To ensure that all interrupts get serviced, poll the Interrupt Identification Register at the end of the interrupt service routine, and service each subsequent interrupt request until the Interrupt ID Register is 'empty.' This may seem a burden, but it actually lowers overall interrupt service time since the normal overhead associated with the additional interrupts (PUSHes, POPs, and IRET) is eliminated.

3. Receiving corrupted data will cause both a 'Received Data' and a 'Received Data Error' interrupt to occur. The Line Status Register and the Receiver Buffer Register must be read to clear this condition.

Although software for interrupt-driven I/O is difficult to write and debug, the results are worth the effort. Table 11 illustrates the execution speeds of the interrupt service routines that are included in the driver listings (listings start on page 17).

Whether you add the interrupt circuits or not, the HB-4 I/O board will offer versatile and powerful serial ports to expand the capabilities of your S-100 system at a nominal cost.

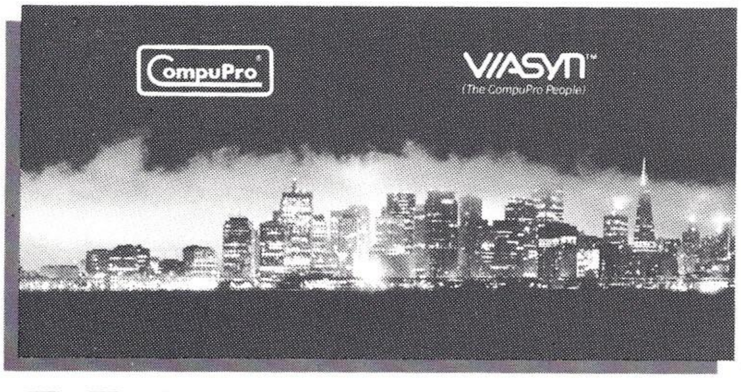
Enjoy the project!

All the parts required for this project can be ordered from most mail-order electronics suppliers for about $130 or less.

Kevin Parker, the author of this article, is considering having printed circuit boards made for those who would like a 'permanent' version of this board. He figures that the cost per PC board would be about $90 if enough people wanted one. This would allow building the 4-port board for about $150.

If you are interested in a printed board, drop us a note (S-100 Journal, PO Box 1914, Orem, UT 84057), and we'll send your name to Kevin.

Multi-User Systems



- Word Processing, Accounting, Database Management Software
- Needs Analysis, System Installation, User Training
- Application Software Design in DATAFLEX
- On-site Service and Support

COCHLIN COMPUTER SYSTEMS, Inc.

330 Townsend Street #107, San Francisco, CA 94107
An Authorized CompuPro System Center

(415) 495-5999

Listing continued from page 21

```
;------------------------> Command Routines <-------------------------------

;       IN_IOCTL - Gets configuration information about a serial port
IN_IOCTL        PROC    NEAR
                CMP     CX,2            ;Make sure there is enough room for the
                JB      NOT_ENOUGH      ;  'input buffer count' word.
            IF  INTERRUPT_IO
                MOV     AX,[BX].IQ_BACK
                CMP     AX,[BX].IQ_FRONT
                JAE     CALC_LEN        ;Apply an adjustment if the buffer 'back'
                ADD     AX,INQ_SIZE     ;  pointer is wrapped around.
      CALC_LEN: SUB     AX,[BX].IQ_FRONT
            ELSE
                MOV     AL,[BX].STATUS
                CBW
            ENDIF
                STOSW
                SUB     CX,2
                CMP     CX,2            ;Ensure enough room for the output buffer
                JB      NOT_ENOUGH      ;  size in the destination buffer.
            IF  INTERRUPT_IO
                MOV     AX,[BX].OQ_BACK     ;Calculate the amount of data in
                CMP     AX,[BX].OQ_FRONT    ;  the output buffer.
                JAE     CALC_LEN1
                ADD     AX,OUTQ_SIZE
     CALC_LEN1: SUB     AX,[BX].OQ_FRONT
            ELSE
                XOR     AX,AX               ;No output buffer for polled I/O.
            ENDIF
                STOSW
                SUB     CX,2
                CMP     CX,2            ;Ensure enough room for the 'baud rate'
                JB      NOT_ENOUGH      ;  word in the destination buffer.
                ADD     DX,3            ;Point to the Line Control Register
                IN      AL,DX
                MOV     SI,AX
                OR      AL,10000000B    ;Setup to read the baudrate divisor
                CLI                     ;No interrupts while reading Baud Divisor
                OUT     DX,AL
                SUB     DX,2            ;Get the MSB of the divisor
                IN      AL,DX
                NOP                     ;Stall for 3 clock cycles
                MOV     BH,AL
                DEC     DX              ;Get the LSB of the divisor
                IN      AL,DX
                NOP                     ;Stall for 3 clock cycles
                MOV     BL,AL
                MOV     AX,SI
                ADD     DX,3            ;Point to Line Control Register and
                OUT     DX,AL           ;  reset the DLA bit.
                STI
                MOV     DX,UPPER_WORD   ;Calculate the baudrate (in bits/sec)
                MOV     AX,LOWER_WORD   ;  based on the 8250 clock frequency
                DIV     BX              ;  and the baudrate divisor.
                STOSW                   ;Save the baud rate in the destination
                SUB     CX,2            ;  buffer.
                JZ      NOT_ENOUGH      ;1 more byte required in dest. buffer
                MOV     AX,SI           ;Recover contents of the line control reg.
                MOV     AH,AL
                AND     AL,00000011B    ;Get the data word length
                ADD     AL,5
                TEST    AH,00000100B    ;Get the number of stop bits
                JZ      ONE_STOP_BIT
                OR      AH,01000000B
                TEST    AH,00000011B
                JZ      ONE_STOP_BIT
                ADD     AL,01000000B
  ONE_STOP_BIT: SHL     AH,1
                AND     AH,00110000B    ;Get parity checking information
                OR      AL,AH
                STOSB                   ;Store the serial line parameters
                DEC     CX              ;  in the destination buffer.
    NOT_ENOUGH: MOV     AX,0100H        ;Function request is complete
                RET
IN_IOCTL        ENDP

;       NON_DES_INP - Returns a character from the device's input buffer
;                     without removing the character from the buffer.
;                     Busy status is returned if no characters are in
;                     the device's input buffer.
NON_DES_INP     PROC    NEAR
            IF  INTERRUPT_IO
                MOV     SI,[BX].IQ_FRONT
                CMP     SI,[BX].IQ_BACK         ;If buffer is empty, return BUSY.
                JE      BUSY_STAT
                LODSB                           ;Otherwise, get the 1st data byte.
            ELSE
                MOV     AL,[BX].BUFFER          ;Is there a valid character in
                CMP     [BX].STATUS,1           ;  the input buffer?
                JAE     STORE                   ;If not, is there a character
                ADD     DX,5                    ;  waiting in the UART?
                IN      AL,DX
```

▶ ED INTERFACE

(continued from page 7)

marketplace. I think, however, that the S-100 community needs to concern itself less with the issues of IBM-compatibility and concentrate more on those areas where S-100 systems have no peer.

Phil Cochlin
President
Cochlin Computer Systems Inc.
San Francisco, California

I could not have said it better myself. Thanks. •*Jay*

Help for Z-System on SBC-100

I run a Sierra Data Sciences SBC-100 on Z-System (ZCPR3 and ZRDOS). I will gladly provide my BIOS modifications to anyone having this board who would like to try ZCPR3.

Keep up the good work.

Michael R. Broschat
2610 E. Aloha
Seattle, WA 98112
(206) 328-0561

Thanks for the cooperative spirit. I've published your full address in case someone wants to take advantage of your offer. •*Jay*

We Listen

I have just finished reading my first issue of your magazine (Vol. 1, No. 3) and I would like to express my dissatisfaction about the way the magazine is put together. It resembles a BASIC program with its GOTO statements. What is so complicated about having an article on continuous pages?

The other thing I hate is to have advertising mixed with technical articles even though this seems to be a standard practice in the industry.

Finally, contrary to your claim about making the reply deck more convenient to request information from advertisers, I think that you are complicating our lives. Having to search through a deck of cards and filling in my address for each request is time consuming. The bingo card works well, why not use it?

I welcome the S-100 Journal since I have an S-100 system myself, but please consider the reader. I hope that you will consider my comments for future issues of the S-100 Journal.

Jean J. Labrosse
San Diego, California

One of the main reasons for breaking articles was for better usage of color pages which fall in specific places in the magazine. However, after reading yours and other similar comments, we gave it a few extra days of thought (and work!) and found that it was not impossible (although it ***is*** *more complicated than you would anticipate) to have whole articles after all. As you can see, in this issue only this column and the program listings are broken. The whole listing wouldn't fit anywhere. But it's not so bad because, if you pull out the binding wires, you'll notice that the listing is together in double pages, perfect for photocopying and making a small booklet.*

We need the ads in the articles to help the text catch up with corresponding figures and tables, among other reasons. Nevertheless, I think you will find them better distributed this time.

Our reply cards were very well received, hence we will continue to publish them. We do not have the resources to process bingo cards (which I think are an inferior system anyhow), so it is between filling out your address or sending a letter.

Thanks for writing. •Jay

Support for Older Systems

Enclosed is my two-year renewal. I find the S-100 Journal a very interesting and valuable publication. Although I have sold articles to other magazines, and have had papers in professional publications of the IEEE, I am the most proud of my article, 'An Editor in BASIC,' that was published in your first issue. The generous terms you offered, and prompt payment, were a pleasant and welcome change from the treatment authors receive from many publications in the computer field. I recommend your magazine both to readers and to authors.

Now, if I may, I would like to make

```
                AND     AL,DATA_AVAIL
                JZ      BUSY_STAT                       ;If so, put it in the buffer.
                SUB     DX,5
                IN      AL,DX
                MOV     [BX].BUFFER,AL
                MOV     [BX].STATUS,1                   ;Set buffer status byte.
            ENDIF
     STORE:     LDS     BX,REQ_HDR_PTR                  ;Put the character in the
                MOV     [BX].RDND_CHAR,AL               ;  Request Header.
     EXIT:      XOR     CX,CX                           ;No bytes were transferred
                MOV     AX,0100H                        ;Command is complete.
                RET
NON_DES_INP     ENDP

;       GET_IN_STAT - Returns serial port input buffer status.
GET_IN_STAT     PROC    NEAR
            IF  INTERRUPT_IO
                MOV     AX,[BX].IQ_FRONT
                CMP     AX,[BX].IQ_BACK                 ;Input requests will have to wait
                JE      BUSY_STAT                       ;  if the input buffer is empty.
            ELSE
                CMP     [BX].STATUS,1                   ;Is there a character in the input
                JAE     EXIT                            ;  buffer?
                ADD     DX,5                            ;If not, is there a character
                IN      AL,DX                           ;  waiting in the UART?
                AND     AL,DATA_AVAIL
                JZ      BUSY_STAT
                SUB     DX,5
                IN      AL,DX                           ;If the UART has a character, get
                MOV     [BX].BUFFER,AL                  ;  it, put it in the input buffer,
                MOV     [BX].STATUS,1                   ;  and update buffer status.
            ENDIF
                XOR     CX,CX                           ;No transfers made
                MOV     AX,0100H                        ;Command is complete
                RET
GET_IN_STAT     ENDP

;       FLUSH_INP_BUF - Flushes input buffer (clears any pending character in the
;                       UART receive buffer for polled I/O).
FLUSH_INP_BUF   PROC    NEAR
            IF  INTERRUPT_IO
                CLI
                MOV     SI,[BX].IQ_FRONT
                MOV     [BX].IQ_BACK,SI
                STI
            ELSE
                MOV     [BX].STATUS,0                   ;Clear buffer status
                IN      AL,DX                           ;Remove character from the UART
            ENDIF
                XOR     CX,CX                           ;No transfers made this command
                MOV     AX,0100H
                RET
FLUSH_INP_BUF   ENDP

     BUSY_STAT: XOR     CX,CX               ;Don't return a byte count for this command
                MOV     AX,0300H            ;No characters in the input buffer
                RET

;       DEV_OUTPUT - Outputs the character string pointed to by ES:SI whose
;                    length is contained in CX.
DEV_OUTPUT      PROC    NEAR
                JCXZ    NO_DATA
                MOV     SI,DI               ;Set up source data buffer pointer
                MOV     AX,ES               ;  registers.
                MOV     DS,AX
            IF  INTERRUPT_IO
                MOV     AX,CS               ;Set up output buffer pointer registers.
                MOV     ES,AX
     SET_LIMIT: MOV     AX,BX                       ;AX points to end of buffer area.
                ADD     AX,OQ_END
    CHEK_FULL1: MOV     DI,CS:[BX].OQ_FRONT             ;If full, wait here until there
                CMP     DI,CS:[BX].OQ_BACK              ;  is room in the buffer.
                JA      NO_ADJ
                ADD     DI,OUTQ_SIZE
        NO_ADJ: SUB     DI,CS:[BX].OQ_BACK
                DEC     DI
                JZ      CHEK_FULL1          ;DI contains amount of room left in buffer.
                SUB     CX,DI               ;Calculate how much data will be loaded
                JNS     NOT_ALL             ;  during this cycle.
                ADD     CX,DI               ;More room is available than required.
                MOV     DI,CX
                XOR     CX,CX
       NOT_ALL: PUSH    CX                  ;Save the amount that will be loaded next.
                MOV     CX,DI               ;Load the count.
                MOV     DI,CS:[BX].OQ_BACK          ;Initialize the buffer pointer.
                CLI                         ;No interrupts while loading buffer.
     LOAD_BUF:
                MOVSB
                CMP     DI,AX               ;Is it time to wrap the pointers?
                LOOPNE  LOAD_BUF            ;Load buffer until wrap around occurs.
                JCXZ    NEXT_LOAD
                SUB     DI,OUTQ_SIZE
                JMP     SHORT LOAD_BUF
     NEXT_LOAD: JNE      SAVE_PTR
                SUB      DI,OUTQ_SIZE           ;Don't let the pointer go out of area!
      SAVE_PTR: MOV      CS:[BX].OQ_BACK,DI  ;Update the End of Buffer pointer
                STI
                POP      CX
                INC      DX                  ;Enable the 'Transmitter Holding Register
                IN       AL,DX               ;  empty' interrupt to start transmitting.
                TEST     AL,0010B            ;Is transmit interrupt already active?
                JNZ      XMITING
                OR       AL,0010B
                NOP                          ;Stall for 3 clocks.
                OUT      DX,AL
       XMITING: DEC      DX
                JCXZ     NO_DATA             ;Return if all data has been loaded into
                JMP      SHORT SET_LIMIT ;  the output buffer.
            ELSE
      PORT_CHK: ADD      DX,5                ;Point to UART status port
      PORT_RDY: IN       AL,DX               ;Wait until UART is ready to transmit.
                AND      AL,XMIT_RDY
                JZ       PORT_RDY
                LODSB                        ;Get the character and send it.
                SUB      DX,5
                OUT      DX,AL
                LOOP     PORT_CHK
            ENDIF
       NO_DATA: MOV      AX,0100H            ;Command is complete
                RET
DEV_OUTPUT      ENDP

;       GET_OUT_STAT - Returns the status of the UART output buffer.
GET_OUT_STAT    PROC     NEAR
            IF  INTERRUPT_IO
                MOV      AX,[BX].OQ_FRONT          ;Calculate the amount of room
                CMP      AX,[BX].OQ_BACK           ;  left in the output buffer.
                JA       DET_LEN
                ADD      AX,OUTQ_SIZE
DET_LEN:        SUB      AX,[BX].OQ_BACK
                DEC      AX
                JNZ      FAST_OUT                  ;Requests for output will not be
                JMP      BUSY_STAT                 ;  fast if output buffer is full.
            ELSE
                ADD      DX,5                ;Point to the UART status register
                IN       AL,DX
                AND      AL,XMIT_RDY
                JNZ      FAST_OUT
                JMP      BUSY_STAT
            ENDIF
      FAST_OUT: MOV      AX,0100H
                XOR      CX,CX
                RET
GET_OUT_STAT    ENDP


            IF  INTERRUPT_IO
;       FLUSH_OUT_BUF - Clears the interrupt-driven device output buffer
FLUSH_OUT_BUF:  MOV      SI,[BX].OQ_BACK
                MOV      [BX].OQ_FRONT,SI
                XOR      CX,CX                   ;No transfers occurred.
                MOV      AX,0100H
                STI
                RET
            ENDIF

;       DEV_INPUT - Transfers CX bytes from the serial port to the destination
;                   buffer pointed to by ES:DI.
DEV_INPUT       PROC     NEAR
                JCXZ     COMPLETE
            IF  INTERRUPT_IO
                MOV      AX,BX
                ADD      AX,IQ_END
                MOV      SI,[BX].IQ_FRONT    ;Unload from the front of the buffer.
       CHEK_MT: CMP      SI,[BX].IQ_BACK     ;Wait until the buffer has data
                JE       CHEK_MT             ;  (it's interrupt driven!)
    UNLOAD_BUF: MOVSB                        ;Get a data byte from the input buffer.
                CMP      SI,AX               ;Is it time to wrap around?
                JNE      CHEK_DONE
                SUB      SI,INQ_SIZE
     CHEK_DONE: MOV      [BX].IQ_FRONT,SI
                LOOP     CHEK_MT             ;Return when we have all of the data.
            ELSE
                CMP      [BX].STATUS,1             ;Get the first byte from the input
                JB       NONE_IN_BUF               ;  buffer if buffer has a character.
                MOV      AL,[BX].BUFFER
                STOSB
                MOV      [BX].STATUS,0             ;Buffer is now empty
                DEC      CX
                JZ       COMPLETE
   NONE_IN_BUF: ADD      DX,5
       CHK_INP: IN       AL,DX                     ;Wait for data to be received.
                AND      AL,DATA_AVAIL
                JZ       CHK_INP
                SUB      DX,5
```

a couple of suggestions. First, how about paying more attention to those who are using an Altair or IMSAI. We have fallen in love with the things you can do with a front panel, and we don't want to give up those lights and switches. Yet, we would like to upgrade, but find little or no hardware or software which is compatible with front panel operation. I am sure many of your readers have found, or devised, solutions to at least some of the problems involved. If you can get some of them to write articles showing how they solved these problems, I, for one, would really appreciate them.

Second, I would like to see more articles on BASIC programs. I don't particularly care what the programs are for, as I seldom copy a program, but I do carefully read published programs looking for 'tricks of the trade.' I refer to things like the trick I explained in my article for using BASIC's EDIT to edit strings stored in a string table. I use it in every program that stores strings in a single or multiple dimension string table. I found a trick in a simple game program in *Family Computing* which I now incorporate in most of the programs I write. Different people's minds work in different ways, and some novice may stumble onto a programming trick that I have never seen in thirty years of programming, and I am not too proud to learn from a novice.

I would also like to see some reviews of BASIC interpreters, those that are usable in S-100 hardware. I would like to find a ROM-based BASIC which has all the features of the MITS BASIC that I now use, but which has the required 'hooks' for installing a DOS. I am a real believer in BASIC, although I also program in assembly language and FORTRAN. Some of the newer BASICs are wonderful, but I don't know of any for S-100 machines. I mean those that allow GOTO and GOSUB to labels and do not require line numbers, permit the use of local and global variables, and support called procedures, both internal and external to the program.

Incidentally, some programmers denigrate BASIC because it is possible to write unstructured programs.

That is not a fault of the language, but a fault of the person using it. I hate spaghetti programs too, but sometimes an unstructured GOTO can save a lot of lines of programming.

Again, keep up the good work. As long as you continue S-100 Journal, and I am able to read, I will be a subscriber.

Viron E. Payne, Sr.
Merritt Island, Florida

When we started publication of S-100 Journal, I promised that we would support both old and new systems. I intend to keep that promise. However, many potential writers of articles about older systems probably think that their articles would be outdated and therefore rejected and so do not submit them. I want to assure everyone that we will consider and publish articles on older systems as well as new. We want S-100 Journal to provide complete information on all that is new in S-100 and even to foster the development of new S-100 products, but we also want to be a practical and useful magazine for every owner of any type of S-100 system.

Although S-100 Journal is mostly hardware-oriented, articles featuring BASIC programs are always welcome, as are articles featuring any other language running on S-100 micros.

Maybe these comments and your letter will encourage authors to submit a few articles on the subjects that you mention. Thanks for your support. •*Jay*

```
                IN      AL,DX                   ;Get the data and store it.
                STOSB
                LOOP    NONE_IN_BUF
            ENDIF
     COMPLETE:  MOV     AX,0100H
                RET
DEV_INPUT       ENDP

            IF  INTERRUPT_IO - 1

;       LIM_INPUT - Reads 1 byte from the UART and returns it in buffer pointed
;                   to by ES:DI.
LIM_INPUT       PROC    NEAR
                JCXZ    NO_DATA2
                DEC     CX                      ;One character will be read.
                CMP     [BX].STATUS,1           ;Get the character from the buffer
                JB      BUF_MT                  ;  if a character is available.
                MOV     AL,[BX].BUFFER
                STOSB
                MOV     [BX].STATUS,0           ;Buffer is now empty.
     NO_DATA2:  MOV     AX,0100H
                RET
      BUF_MT:   ADD     DX,5                    ;Otherwise, wait for the data
    CHAR_AVAIL: IN      AL,DX                   ;  from the UART.
                AND     AL,DATA_AVAIL
                JZ      CHAR_AVAIL
                SUB     DX,5
                IN      AL,DX
                STOSB                           ;Get the data and save it.
                MOV     AX,0100H
                RET
LIM_INPUT       ENDP

            ENDIF

;       OUT_IOCTL - Outputs a serial port configuration string (up to 64K in
;                   length) to the UART.
                LEGAL_COM  DB  "BFPDS"          ;Allowable I/O control commands.
                 COM_TBLE  DW  CHG_BAUD
                           DW  FLUSH
                           DW  CHG_PARITY
                           DW  CHG_WORD_LEN
                           DW  CHG_STOP_BIT
            PORT_HDR_SAVE  DW  ?
OUT_IOCTL       PROC    NEAR
                JCXZ    NO_COM
                MOV     PORT_HDR_SAVE,BX
                MOV     SI,DI               ;Line up segment registers and pointers.
                MOV     AX,ES
                MOV     DS,AX
                MOV     AX,CS
                MOV     ES,AX
     PARSE_COM: MOV     DI,OFFSET LEGAL_COM
                PUSH    CX                      ;Determine next type of command
                MOV     CX,5                    ;  string to be processed.
                LODSB
                REPNE   SCASB
                POP     CX
                JNE     ERROR                   ;Return to system if unknown command
                SUB     DI,OFFSET LEGAL_COM + 1
                SHL     DI,1
                CALL    CS:COM_TBLE[DI]         ;Execute the command
                LOOP    PARSE_COM               ;Get the next command
        NO_COM: MOV     AX,0100H
                RET
OUT_IOCTL       ENDP

;       Baud rate setting routine.
 CHG_BAUD:      CMP     WORD PTR [SI],4         ;Excessively low baudrates will
                JB      FAULT                   ;  cause divide errors.
                CMP     CX,3                    ;Ensure source buffer contains the
                JB      ERROR1                  ;  entire command.
                CMP     WORD PTR [SI],56000     ;Baudrates are limited to 56k baud
                JA      FAULT
                CLI             ;No interrupts while accessing the Baud Divisor.
                ADD     DX,3                    ;Set up to access the baudrate
                IN      AL,DX                   ;  divisor register
                MOV     BL,AL
                OR      AL,80H
                OUT     DX,AL
                SUB     DX,3                    ;Determine the baudrate divisor
                MOV     DI,DX                   ;  and load it into the Baudrate
                MOV     DX,UPPER_WORD           ;  Divisor register (in the UART).
                MOV     AX,LOWER_WORD
                DIV     WORD PTR [SI]
                MOV     DX,DI
                OUT     DX,AL
                ADD     DX,1
                MOV     AL,AH
                OUT     DX,AL
                MOV     AL,BL                   ;Restore the UART Line Control
                ADD     DX,2                    ;  Register to its previous value.
                OUT     DX,AL
                STI
                ADD     SI,2                            ;Point to next command string.
                SUB     CX,2
                SUB     DX,3                            ;Point to base address of the UART.
                RET

;       Parity setting routine.
CHG_PARITY:     CMP     CX,1
                JE      ERROR1
                ADD     DX,3                            ;Point to Line Control Register.
                IN      AL,DX
                AND     AL,11100111B                    ;Mask off the parity control bits
                CMP     BYTE PTR [SI],"N"               ;  and modify the Line Control
                JE      PUT_PARITY                      ;  Register image according to
                OR      AL,00001000B                    ;  the requested parity parameters.
                CMP     BYTE PTR [SI],"O"
                JE      PUT_PARITY
                CMP     BYTE PTR [SI],"E"
                JNE     ERROR1
                OR      AL,00010000B
    PUT_PARITY: OUT     DX,AL                   ;Update the Line Control Register.
                INC     SI                      ;Point to the next command string.
                SUB     DX,3
                DEC     CX
                RET

       ERROR1:  ADD     SP,2
        ERROR:  MOV     AX,8103H                        ;Return an Unknown Command error
                RET                                     ;  to the operating system.
       FAULT:   MOV     AX,810CH                        ;Return a General Failure error
                ADD     SP,2                            ;  to the operating system (due
                RET                                     ;  to a parameter out of range).

;       Data word length setting routine.
 CHG_WORD_LEN:  CMP     CX,1                    ;Ensure all required data is present.
                JE      ERROR1
                MOV     BL,[SI]
                SUB     BL,"5"                  ;Convert the requested length to a
                CMP     BL,4                    ;  binary number.
                JAE     FAULT
                ADD     DX,3                    ;Update the Line Control Register with the
                IN      AL,DX                   ;  new data word length.
                AND     AL,11111100B
                OR      AL,BL
                OUT     DX,AL
                SUB     DX,3
                INC     SI                      ;Point to next command string.
                DEC     CX
                RET

;       Stop bit selection routine.
 CHG_STOP_BIT:  CMP     CX,1                    ;Ensure all required data is present
                JE      ERROR1
                MOV     BL,[SI]
                SUB     BL,"1"                  ;Determine desired setting
                CMP     BL,1
                JA      FAULT
                SHL     BL,1
                SHL     BL,1
                ADD     DX,3
                IN      AL,DX
                AND     AL,11111011B
                OR      AL,BL
                OUT     DX,AL
                SUB     DX,3
                INC     SI                      ;Next command string
                DEC     CX
                RET

;       Device I/O buffer flushing routine.
FLUSH:          MOV     BX,CS:PORT_HDR_SAVE
                CMP     CX,1
                JE      ERROR1
                LODSB                           ;Determine which buffer to flush
                CMP     AL,"I"
                JE      INP_FLUSH
                CMP     AL,"O"
                JNE     ERROR1
            IF  INTERRUPT_IO
                ADD     BX,OQ_FRONT - IQ_FRONT          ;Point to correct buffer
            ENDIF
INP_FLUSH:
            IF  INTERRUPT_IO
                CLI
                MOV     AX,CS:[BX].IQ_FRONT                     ;Flush the buffer.
                MOV     CS:[BX].IQ_BACK,AX
                STI
            ELSE
                MOV     BYTE PTR CS:[BX].STATUS,0               ;Clear the input buffer
                IN      AL,DX
            ENDIF
```

bits

*The Bits department is for publishing **non-commercial** small advertisements. There is **no charge** for subscribers to place an ad. However, please limit your message to **50 words.** If your ad has more words, there is a charge of $1.00 for each word over 50.*

You may take advantage of this service to sell or buy used S-100 hardware, trade personal programs, etc. Send ads and prepayment (if applicable) to S-100 Journal, BITS, PO Box 1914, Orem, UT 84057.

FOR SALE

56 Mb hard disk subsystem with Konan controller as used in Gifford S-100 System, $2,000. SMD to S-100 Hard Disk Konan controller, any reasonable offer. John S. Moseid, 805-499-5780.

Teletek Systemaster SBC w/CPM, $395. ParaGraphic video/graphics board: to 512×576, H19 emulation, w/software, $295. Calif. Dig. 256K RAM, $150. Sequential Circuits M206 Musicboard: 6-note polyphonic, 80 preset voices, 20 user-created, serial interface, $195. Above complete system in Integrand mainframe, w/Wordstar, Cnix, Macro-80, Aztec C, UO-Lisp, etc., $995. Also, HSC Motorola 68000 board w/768K RAM, Z80 I/O interface, w/CPM68K, C, etc., $795. Prof. Buckley, 603-868-5006.

CompuPro 20-slot S-100 enclosure/motherboard, Paradynamics model 2200D dual 8″ enclosure, 2 Mitsubishi model m-2894-63 8″ drives, Advanced Digital 4MHz Z-80 SBC, US Robotics S-100 1200B modem, ADDS model VWPNT terminal, Comrex CR-1 Daisywheel printer, CP/M 2.2 and SCP/80 front end, Pascal MT+ and Aztec C compilers, 100 8″ disks, 2 8″ flip'n'files. Running system, used less than 200 hrs. Package $2000. W, 804-253-4006, H, 804-693-5897.

►

Two NEW S-100 Mainframes by International Instrumentation Incorporated. 12-Slot motherboard. S-100 Switcher power supply. Separate internal power supply for two full height 5¼ floppies. Room to mount floppies internally. Never used. Full documentation including schematics for power supply included. $700 for both. Contact Ira Goodberg 213-650-6327.

Three Datagraphix keyboards with documentation, one with crude case $50 each. PMMI MM-103 S-100 modem with documentation, $50. Gerald Zuckier, 112 Canner St., New Haven, CT 06511, 203-624-0940.

Teletek FDC1 CPY with 2-serial/2-parallel ports; supports popular 5″ and 8″ formats. CompuPro RAM16 64K. 2 Tandon TM848-2 1.2 Mb floppy drives. 1 California Computer System model 2719-I/O 2-parallel/2-serial ports. 2 new unformatted Shugart Associate SA1002 5Mb 8″ hard drives. 1 Hayes Smart 1200 Modem. Integrand Mainframe. Qume QVTY-102 green terminal with detached keyboard. COBOL/FORTRAN/TurboPascal/BASIC. Dystan 8″ alignment disk. Wordstar/Spellstar/Mailmerge/The Word/Webster's Electronic Thesaurus. CP/M operating system with many utilities. Complete cable set, comprehensive documentation, reference material. No reasonable offer refused for the complete package. Call evenings 916-363-8144.

Morrow Decision I S-100 system with: Morrow CPZ80 4MHz CPU card, 65K RAM card, DJ/D2B 8″ floppy controller card, HDC3A 8″/14″ hard disk controller card, second cabinet w/one SS-DD 8″ floppy drive and power supply for 10/15/20Mb 8″ hard drive (drive not incl), one Shugart 14″ 26Mb hard disk drive, power supply, separate cabinet (3 enclosures total). Bootable CP/M 2.2 system included, many diskettes of Morrow BIOS configs, ZCPR3, remote utilites, and a complete InfoSoft Multi I/OS multiuser operating system in source and .REL form (never implemented) on 8-inch diskettes. System is all Morrow. Asking $800 for everything. You pay shipping FOB Tampa, FL. Steve Sanders, 813-791-1938.

►

```
                DEC     CX
                RET                                             ;

;---------------------------> DATA <---------------------------------

                EVEN

REQ_HDR_PTR     DD      ?       ;Pointer to the command header is stored here.

;       Structure of the Request Header passed to driver by the system:
REQ_HDR         STRUC
    HDR_LEN     DB      ?
                DB      ?
    COM_CODE    DB      ?       ;Command to be executed (0 - 12)
    COM_STAT    DW      ?       ;Status of command (returned by driver)
                DB  8 DUP (?)
    RDND_CHAR   DB      ?       ;Character returned during a non-destructive read
    XFER_ADDR   DW      ?,?     ;Transfer address
    BYTE_COUNT  DW      ?       ;Number of bytes to be transferred
REQ_HDR         ENDS

;       Structure of the command list used by each serial device:
COMMANDS        STRUC
                DW      EXIT
                DW      EXIT, EXIT, IN_IOCTL
     COM5       DW      DEV_INPUT
                DW      NON_DES_INP, GET_IN_STAT, FLUSH_INP_BUF, DEV_OUTPUT
                DW      EXIT, GET_OUT_STAT
     COM12      DW      EXIT
                DW      OUT_IOCTL
     PORT_BASE  DW      BASE
COMMANDS        ENDS

        IF   INTERRUPT_IO

;       Structure of data buffer packet used by the interrupt driven I/O system:
BUFFERS         STRUC
    IQ_FRONT    DW      ?                       ;Pointer to 1st byte in input buffer
    IQ_BACK     DW      ?                       ;Pointer to next empty slot in buffer
    IN_Q        DB      INQ_SIZE DUP (?)        ;Input buffer
    OQ_FRONT    DW      ?                       ;Pointer to first byte in output buffer
    OQ_BACK     DW      ?                       ;Pointer to next empty slot in buffer
    OUT_Q       DB      OUTQ_SIZE DUP (?)       ;Output buffer
BUFFERS         ENDS

IQ_END  EQU     IN_Q + INQ_SIZE
OQ_END  EQU     OUT_Q + OUTQ_SIZE

PRNB      BUFFERS     <PRNB.IN_Q,PRNB.IN_Q,,PRNB.OUT_Q,PRNB.OUT_Q>
PRN_COM   COMMANDS    <SHORT_INIT,,,,,FLUSH_OUT_BUF,,>
EPB       BUFFERS     <EPB.IN_Q,EPB.IN_Q,,EPB.OUT_Q,EPB.OUT_Q>
EP_COM    COMMANDS    <SHORT_INIT,,,,,FLUSH_OUT_BUF,,BASE + 8>
MODB      BUFFERS     <MODB.IN_Q,MODB.IN_Q,,MODB.OUT_Q,MODB.OUT_Q>
MOD_COM   COMMANDS    <SHORT_INIT,,,,,FLUSH_OUT_BUF,,BASE  + 16>
AUXB      BUFFERS     <AUXB.IN_Q,AUXB.IN_Q,,AUXB.OUT_Q,AUXB.OUT_Q>
AUX_COM   COMMANDS    <INITIALIZE,,,,,FLUSH_OUT_BUF,,BASE + 24>

        ELSE
;       Structure of data buffer pack used by polled I/O system (no output buffer
;       is necessary):
BUFFERS         STRUC
        BUFFER  DB      ?
        STATUS  DB      0
BUFFERS         ENDS

PRNB      BUFFERS     <>
PRN_COM   COMMANDS    <SHORT_INIT,,LIM_INPUT,,,,,>
EPB       BUFFERS     <>
EP_COM    COMMANDS    <SHORT_INIT,,LIM_INPUT,,,,,BASE + 8>
MODB      BUFFERS     <>
MOD_COM   COMMANDS    <SHORT_INIT,,,,,,,BASE + 16>
AUXB      BUFFERS     <>
AUX_COM   COMMANDS    <INITIALIZE,,,,,,,BASE + 24>

        ENDIF

BLEN    EQU     SIZE  BUFFERS                                   ;

;----------------------> Board Initialization <--------------------------------

;       All code and data below this point are used only during device
;       driver installation and are discarded after use.

SHORT_INIT:     LDS     BX,REQ_HDR_PTR
                MOV     [BX].RDND_CHAR,1                ;1 unit
                MOV     [BX].XFER_ADDR,OFFSET SHORT_INIT
                MOV     [BX].XFER_ADDR + 2,CS
                MOV     AX,0100H                        ;Command is completed
                XOR     CX,CX                           ;No data was transferred
                RET
```

```
;       The following code is executed only once during device initialization
;       to initialize the 4 serial port board.

;       Initialize the serial ports:
INITIALIZE:     CLI                     ;No interrupts during initialization
                MOV     DX,BASE + 3
                MOV     SI,OFFSET INIT_TABLE
                MOV     CX,4                    ;4 ports will be initialized
    INIT_PORT:  MOV     AL,80H          ;Access the baudrate divisor register
                OUT     DX,AL
                SUB     DX,3
                LODSB
                OUT     DX,AL           ;Load LSB of baudrate divisor
                INC     DX
                LODSB
                OUT     DX,AL           ;Load MSB of baudrate divisor
                ADD     DX,2
                LODSB
                OUT     DX,AL           ;Load the line control register
                ADD     DX,8            ;Point to next UART
                LOOP    INIT_PORT

            IF INTERRUPT_IO
;       Initialize the slave 8259 and re-initialize the system master 8259 so that
;       the master will accept interrupts from the slave.
                LODSB
                MOV     DX,SLAVE_59_0
                OUT     DX,AL                   ;Initialize the slave 8259.
                MOV     DX,SLAVE_59_1
                MOV     CX,4
    SLAVE_INIT: LODSB
                OUT     DX,AL
                LOOP    SLAVE_INIT
                MOV     DX,MASTER_59_0
                LODSB
                OUT     DX,AL                   ;Re-initialize the master 8259.
                INC     DX
                MOV     CX,4
   MASTER_INIT: LODSB
                OUT     DX,AL
                LOOP    MASTER_INIT
        ; Set up the interrupt vectors in the Interrupt Table.
                XOR     AX,AX           ;Set up to access the Interrupt Vector Table.
                MOV     ES,AX
                MOV     DI,(40H SHL 2)  ;Slave is in vectors 40H - 47H
                MOV     CX,8
                REP     MOVSW
                MOV     CX,4
                MOV     AL,0101B
                MOV     DX,BASE + 1
       EN_INT:  OUT     DX,AL           ;Enable all 8250 rcvr-generated interrupts
                ADD     DX,8
                LOOP    EN_INT
            ENDIF
                CALL    SHORT_INIT      ;Set the address following the device driver
                STI                     ;  code and data.
                RET

;       Set the individual serial port parameters here.  The first two bytes
;       of each 3-byte sequence is the baud rate divisor.  The third byte of
;       each 3-byte sequence is the Line Control parameter specification byte.
;       Baud rates are entered in 10s of baud (10 ==> 100 baud).
INIT_TABLE      DW  CLOCK / 120         ;Printer is 1200 baud
                DB  03                          ;1 stop bit, no parity, 8 bit data
                DW  CLOCK / 960         ;EPROM burner at 9600 baud
                DB  03
                DW  CLOCK / 30          ;MODEM operates at 300 baud
                DB  03
                DW  CLOCK / 120         ;AUX1 is a spare 1200 baud port
                DB  03

            IF  INTERRUPT_IO

;       8259 Priority Interrupt Controller chip initialization table:
INT_INIT_TABLE: DB      19H,40H,02H,0BH,0F0H    ;Slave 8259 - cascade mode, level
                                                ; triggered, cascade mode, 4 init
        ; words, slave #2, interrrupt vectors 40h - 47h, 8086 mode, buffered mode.

                DB      19H,10H,06H,0FH,0F9H    ;Master 8259 - same as above except
                                                ; master mode with 2 slaves connected
        ; to IR lines 1 & 2 (line 1 has the slave 8259 on the SCP CPU support card,
        ; line 2 has the slave 8259 on the HB-4 board), interrupt vectors 10h - 17h.

;       Interrupt vector initialization table:
VECT_INIT_TABLE:  DW    AUX1_INT_RTN,DEV_CODE
                  DW    MODEM_INT_RTN,DEV_CODE
                  DW    EPROM_INT_RTN,DEV_CODE
                  DW    PRN_INT_RTN,DEV_CODE
            ENDIF

DEV_CODE        ENDS
                END
```

— end of HB-4 driver listings —

S-100 system Lomas Data Products boards, MS-DOS 2.1, 640K, 2 ea. serial & parallel ports, 2 ea. 1.2 Mbyte 8″ drives, 1 DSDD 5¼″ drive, ADM-2 CRT, all manuals, cables, and some spare boards, 300-baud smartmodem, $550. Dan Pritchard, 4721 Bali Ct, Albuquerque, NM 87111, 505-293-5297.

Cromemco 68000/Z80 multiuser system. Cromix/plus operating system, w/ C, Pascal, and BASIC. Runs CP/M applications. Contact Robert Coats, 919-541-9287.

Complete multiuser S-100, 7 slot, (Z-80 CPU) integrated TurboDOS system, 45Mb Micropolis hard-disk drive, 2 8″ floppies, 4 TV-912 terminals, considerable Micropro software. Asking $6,800 (2nd 45Mb drive available). 404-587-0412.

New Fulcrum Omnidisk controller with CP/M, $195. The following are offered as is: Hayes S-100 Micromodem, $45; Vetor Graphic 48K RAM and Z80 boards, $20 each. Gary Van Cott, PO Box 1879, Grafton, VA 23692. 804-898-3680.

Dual Systems 83/80, 12 user system. 1-Mb RAM, 80-Mb hard disk, 8″ floppy, Unix System V, Informix and Unify DBMS, lex WP, two Liberty terminals (one with modem), national service contract, all in excellent condition. Make offer. Tony Hill, 1436 White St., Ann Arbor, MI 48104. 313-761-1736 or 313-665-9773

HARDWARE WANTED

I am interested in purchasing a Technical Design Labs dynamic RAM board with any amount of memory installed and a TDL DDDC disk controller board. Both must be in good working order. Call or write William Leonard, 14956 Ronda Drive, San Jose, CA 95124. 408-377-8582, evenings.

32 BITS ON THE S-100 BUS

Dion Vaughn

The S-100 bus has seen quite a number of single-board and other computers come and go. It has been expanded and upgraded to the point that it is now capable of supporting a microcomputer system of the highest performance.

In this article, I propose a new upgrade that will allow the S-100 bus to support the new 32-bit processors.

We now have a lot of S-100 cards designed to run in 8-bit and 16-bit systems. We want to be able to continue using these boards, not only for economic reasons, but also because most data transfers between processors and peripherals are done 8 bits at a time. In other words, we would not always need an I/O board with 32 data bits. Of course, once a 32-bit bus is established and a 32-bit processor installed, I/O boards could be built with 32-bit data paths from the bus to on-board buffers. The data would then be accepted or sent by the peripheral, at its own rate, through these buffers.

Editor's Note:

With the articles in this issue on 32-bit implementation, we hope to initiate a discussion on this subject.

If you have any good ideas or comments on using 32-bit processors on the S-100 bus, you are encouraged to either submit articles or send us a letter with your comments or solutions.

You may want to use this first article as a starting point for the discussion. Send all materials to:

32-Bit Forum, S-100 Journal
PO Box 1914, Orem, UT 84057.

A successful upgrade must therefore be **compatible** with existing circuit boards while taking advantage of the new performance derived from the 32-bit processors.

One of the greatest performance boosters to be obtained from the new processors is increased clock speed, typically 12-16 MHz or more. However, the S-100 bus was originally designed around a 2-MHz processor. If we implement all the signals the way they are currently used by many S-100 circuit boards, the bus will not be capable of running at 12-16 MHz.

IEEE approved standards for the timing of the bus signals in a very generalized form. One of the main considerations was the many different circuit boards already in use. These boards interpreted and used the bus signals in several different ways. To allow some of the boards to run together properly, delays had to be built in and a 'status valid' signal added. This was particularly important for dynamic RAM boards.

Therefore, some compromises in performance may have to be made if the bus signals continue to be used in the same way. The timing of these signals is such that they simply cannot be driven at the system clock speeds offered by the new processors.

For the new upgrade to maintain compatibility with existing boards, it should provide a new method of using some of the S-100 bus signals without changing any of them. The new method must allow new-technology circuit boards to run at much higher speeds.

SOME ALTERNATIVES

The most efficient way to implement 32 bits on the S-100 bus would probably be to redefine the bus. There are 91 signal lines available. The other ►

Dion Vaughn is a Computer Design Engineer and is president of Magnum Digital. Dion is experienced in the design of industrial controllers and S-100 boards.

9 are power and ground. If 32 lines were used for the address bus, 32 for data, 4 lines for arbitration, and 8 for interrupts, there would still be 15 lines left for control and status signals.

Redefining the bus would offer the advantage of enough signal lines being available on a single backplane.

However, the disadvantages associated with this alternative are very obvious. We would need to have two S-100 mainframes around all the time — one for the 8/16-bit world with its multitude of existing cards, and one for the 32-bit world. Only the mainframe of the present S-100 system could be used with the new 32-bit circuit boards. No existing S-100 circuit boards could be used.

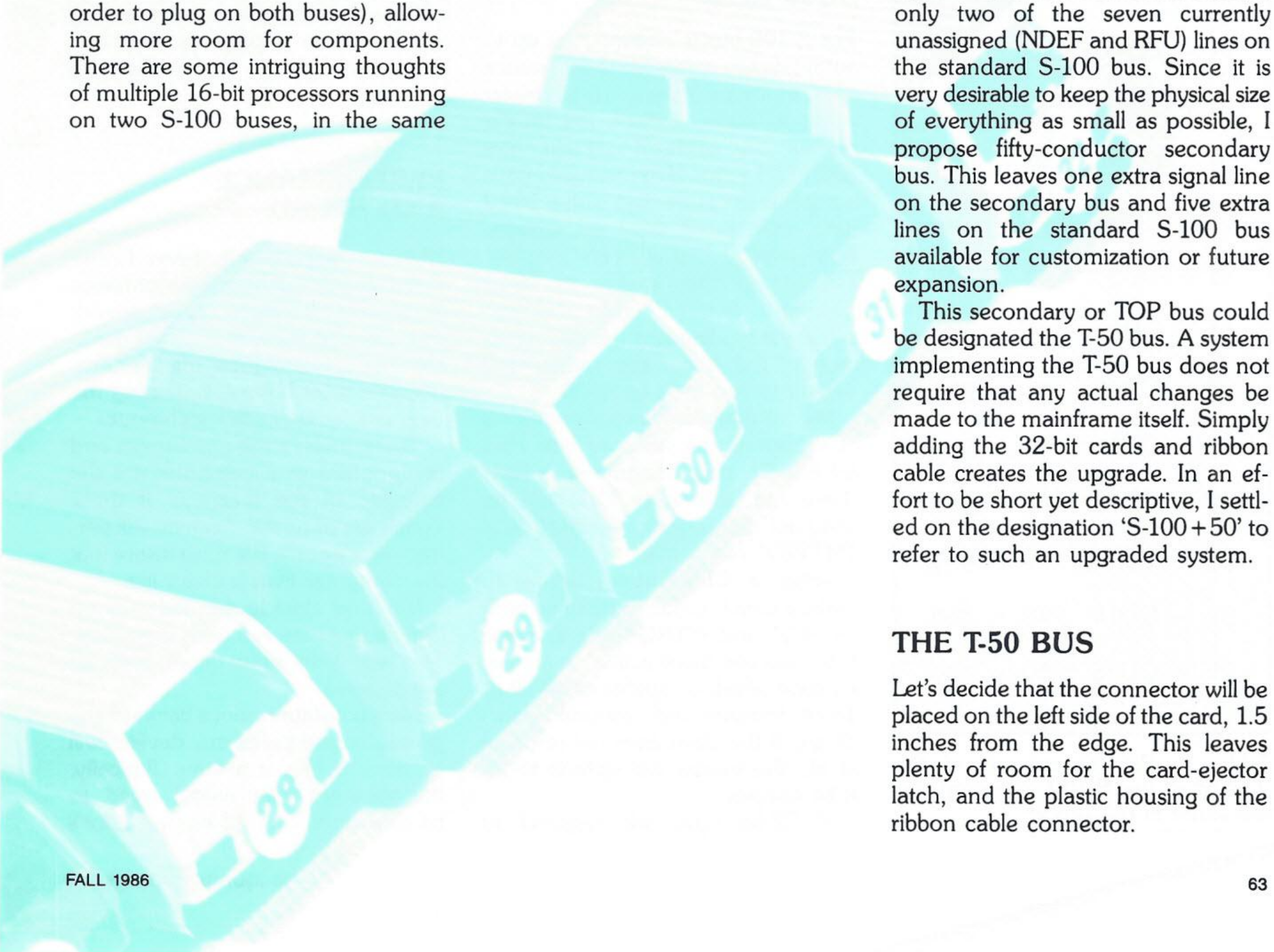
We could make the mainframe physically wider and install two motherboards in it. Two standard S-100 motherboards could be used, or one standard S-100 and one 44-pin bus, or an S-100 and a 50 pin bus, etc.

The advantages of this scheme would be that current circuit boards could still be used in the standard S-100 portion of the mainframe. The new 32-bit cards would be wider (in order to plug on both buses), allowing more room for components. There are some intriguing thoughts of multiple 16-bit processors running on two S-100 buses, in the same mainframe, with one double-wide 32-bit processor card controlling the whole thing.

The disadvantages include the fact that we could not use current mainframes since the new mainframe would be much larger. And problems would arise when trying to adjust card guides, and other hardware inside the mainframe, for the varying widths of circuit boards.

The last, and possibly most feasible, idea for a 32-bit upgrade is to use the S-100 mainframe as it is and to implement the additional address and data lines as a secondary bus on top of the circuit cards. The secondary bus would only be connected to those cards capable of using it. To run at the higher speeds, all future circuit boards could be designed to use the S-100 control and status signals in a slightly different manner.

With this implementation, we can still use all our old circuit boards, we don't have to change the physical size of the system, and we can run the new, higher-performance boards without any limitations. This is the scheme discussed in this article.



32-BIT IMPLEMENTATION

Let's look at the secondary bus first. We need eight more address lines, sixteen more data lines, and at least two control/status lines. The control/status lines will be implemented on the standard S-100 bus. (These two lines will be discussed later). This means that we need an absolute minimum of twenty-four lines on our secondary bus.

Having a ground line next to a signal line causes more capacitance for the signal line driver to overcome. However, having one signal line next to another signal line creates added capacitance plus the additional problem of induced signal noise. Thus, the best method is to use a ground line between each of the signal lines. This means that we now need forty-eight lines in our secondary bus.

Fifty-conductor ribbon cable, IDC connectors, card-edge connectors, etc. are standard and readily available from many suppliers. The next 'standard-size' ribbon cable and connectors would be sixty-conductor. The upgrade discussed here will use forty-eight lines on the secondary bus and only two of the seven currently unassigned (NDEF and RFU) lines on the standard S-100 bus. Since it is very desirable to keep the physical size of everything as small as possible, I propose fifty-conductor secondary bus. This leaves one extra signal line on the secondary bus and five extra lines on the standard S-100 bus available for customization or future expansion.

This secondary or TOP bus could be designated the T-50 bus. A system implementing the T-50 bus does not require that any actual changes be made to the mainframe itself. Simply adding the 32-bit cards and ribbon cable creates the upgrade. In an effort to be short yet descriptive, I settled on the designation 'S-100+50' to refer to such an upgraded system.

THE T-50 BUS

Let's decide that the connector will be placed on the left side of the card, 1.5 inches from the edge. This leaves plenty of room for the card-ejector latch, and the plastic housing of the ribbon cable connector.

THE T-50 BUS

1 – A24	26 – gnd
2 – A25	27 – gnd
3 – A26	28 – gnd
4 – A27	29 – gnd
5 – A28	30 – gnd
6 – A29	31 – gnd
7 – A30	32 – gnd
8 – A31	33 – gnd
9 – NDEF	34 – gnd
10 – ED8	35 – gnd
11 – ED9	36 – gnd
12 – ED10	37 – gnd
13 – ED11	38 – gnd
14 – ED12	39 – gnd
15 – ED13	40 – gnd
16 – ED14	41 – gnd
17 – ED15	42 – gnd
18 – OD8	43 – gnd
19 – OD9	44 – gnd
20 – OD10	45 – gnd
21 – OD11	46 – gnd
22 – OD12	47 – gnd
23 – OD13	48 – gnd
24 – OD14	49 – gnd
25 – OD15	50 – gnd

Table 1. *Pin assignments of the T-50 Bus, the S-100 extension bus discussed in this article.*

In laying out a circuit board, it seems logical to put the new address lines on the left. This allows the installation of address decoders and switches at a location other than in the middle of logic devices that will probably need the data lines. The extra NDEF signal line should be next, followed by the sixteen high-order data bits.

The T-50 data bus is set up like the primary (S-100) data bus. The low-order 8 bits of the high-order data word are accessed by even addresses, and the high-order 8 bits are accessed by odd addresses.

The pin numbering of the T-50 bus (Table 1) is also like that of the S-100 bus. Pin 1 is found at the front left position, pin 2 immediately to the right of pin 1, and so on until pin 25 at the front right position. Pin 50 is just behind pin 25 in the right-rear position, pin 49 immediately to the left of pin 50, and so on until pin 26 just behind pin 1 at the left-rear position.

SIGNAL ASSIGNMENTS ON THE STANDARD S-100 BUS

The S-100 bus is currently set up to automatically recognize the existance of 16-bit slaves. When a 16-bit master wants to access a slave, the master makes the sixteen-request line (sXTRQ*) active. If the accessed slave is capable of responding with a 16-bit data path, it will drive the sixteen-acknowledge line (SIXTN*) active. The master then makes its 16-bit transfer. But, if the sixteen-acknowledge line was not driven active by the slave, the master will default to two 8-bit transfers.

We will make only two assignments of signals on the standard bus. Pins 69 and 71, now designated as RFU (Reserved for Future Use) will be assigned the signals **sTTRQ*** and **THTWO*** respectively.

When a 32-bit master desires to make a data transfer, it will drive both sXTRQ* and sTTRQ* active. Since both lines are driven active, an accessed slave which is capable of 32-bit or 16-bit transfers will respond accordingly. If the slave does not respond at all, the master will default to an 8-bit transfer.

A 32-bit slave will respond to sXTRQ* and sTTRQ* by driving THTWO* active. If the slave is capable of 32-bit or 16-bit transfers (with buffers or 16-bit word latches), and is accessed with only sXTRQ* active by a 16-bit master, it will respond by driving SIXTN* active.

TEMPORARY MASTERS

When a system has more than one processor, one of them is normally configured as a master and the others as temporary masters. Another method is to have all processors configured as temporary masters and a bus arbitrator controlling them. Either way, when a temporary master has the bus, it can legally access any unprotected memory or I/O ports that any other processor can.

Some of the new 32-bit processors have a linear address range of 4 gigabytes. Less powerful processors normally have an address range of 64K to 16 megabytes. Therefore, for any processor to access any memory or I/O port outside of its linear range, software/firmware must first cause the higher-capacity processor to latch the address such that the desired memory will be within a page or bank addressable by the lower-capacity processor.

PERFORMANCE AND SPEED

Higher speed normally means higher performance, but higher performance is required to obtain higher speed. Therefore, to run at the highest possible speed, the highest quality components must be installed using the very best construction techniques.

The signals in the conductors and components of a computer are the very life of the machine. If these signals are dirty, the machine will perform very poorly. We must insure that the computer lives a clean life.

We have already decided to put a ground line between each signal line. This will help hold down induced signal noise.

Most communications between the processor and peripheral devices will be done 8 bits at a time. Typically, the modules that would need to be connected to a 32-bit processor's extended address and data buses would be memory and coprocessors. Most of the circuit boards in a typical system will not need to be connected to the T-50 bus. Thus, the T-50 bus can be kept relatively short which will also help hold down the noise that might be generated on a longer bus.

Termination requirements will depend mostly on clock speed and physical length of the bus. Active termination could be provided on a bus-arbitrator card, the processor, or any other functional circuit board connected to the T-50 bus.

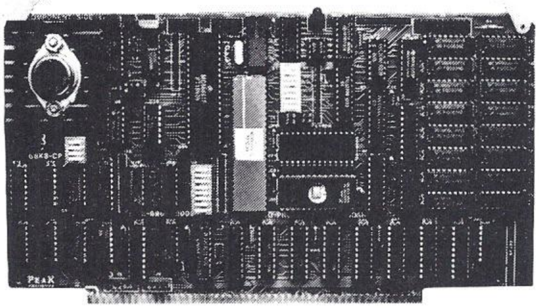
The next consideration is signal timing. As already mentioned, not all of the generalized S-100 signals can be driven to the high speeds that we are now considering. Several of the bus signals were specified with built-in delays so that older, already available circuit boards could be used. These delays also facilitated the use of the bus signals as strobes for clocking data onto and off various modules.

Currently, the bus signals pWR* and pDBIN are the generalized write and read strobes respectively. However, using more sophisticated components now available, circuit boards can be designed that implement their own strobes and respond to accesses very quickly. This would allow the processor to run at full speed with all cards that are capable of keeping up. Of course lower-performance parts, or those designed to use the delayed signals, could still be used by generating any necessary wait states.

Figures 1 and 2 (on next page) show methods that I have used for generating on-board strobes.

Figure 1 is the simplest method and works very well. Assuming a general SELECT* signal produced when the board is addressed, we simply use the trailing edge of this SELECT* signal to clock in the data. This allows the components on the board to use the entire bus cycle to get themselves set up (satisfy hold time requirements). Our main consideration here is to be certain that the data to be latched in is still valid when the SELECT* line goes false — this is when the data will be clocked in.

Figure 2 shows a method which will allow a specific time, depending on clock periods, before data is clocked in. We simply use one or more



68K8-CP

Expand Your System with a 68000 CoProcessor

Peak Electronics' 68K8-CP is a high performance 68000 software development package designed to easily integrate into your existing S-100 system. The package consists of the 68K8-CP coprocessor card, CP/M-68K, and a software toolkit that includes a UNIX V7 compatible floating point C compiler and a symbolic debugger.

Any system running CP/M®-2.2, CP/M-3.0 or CP/M-86 can be running CP/M-68K within minutes without any change in existing hardware or software. This card does not replace your current processor. All of the original system's devices (RAM, disks, and other peripherals) are immediately available to the user of CP/M-68K. All files can be accessed by whichever operating system is currently active. Control is transferred between operating systems with a simple one line command.

Features:

- Does not replace your current CPU card or software
- Includes CP/M-68K with UNIX® V7 compatible floating point C compiler and a symbolic debugger
- All developed C and Assembly code is fully relocatable and ROMable
- 8 or 10Mhz CPU with no wait state RAM
- 128K bytes of RAM expandable to 512K
- 2 serial and 1 parallel I/O ports
- IEEE-696-1983, S-100 Compatible
- 30 day money back guarantee
- 1 year parts and labor warranty

Complete Package: $995.00

VISA or Master Card Accepted

PEAK
Electronics
P.O. Box 700112, San Jose, CA 95170-0112
(408) 253-5108

SUBSCRIBE

To S-100 Journal

YES! Please start my subscription to S-100 Journal.

☐ $14.00 for 1 year (4 issues) ☐ $25.00 for 2 years (8 issues)

(Canada and Mexico, add $3.00 per year. Other countries add $6.00 per year.)

☐ Bill me (U.S. only) ☐ My Payment is enclosed (1 extra issue free)

NAME: ______________________________

ADDRESS: ______________________________

____

FALL 86

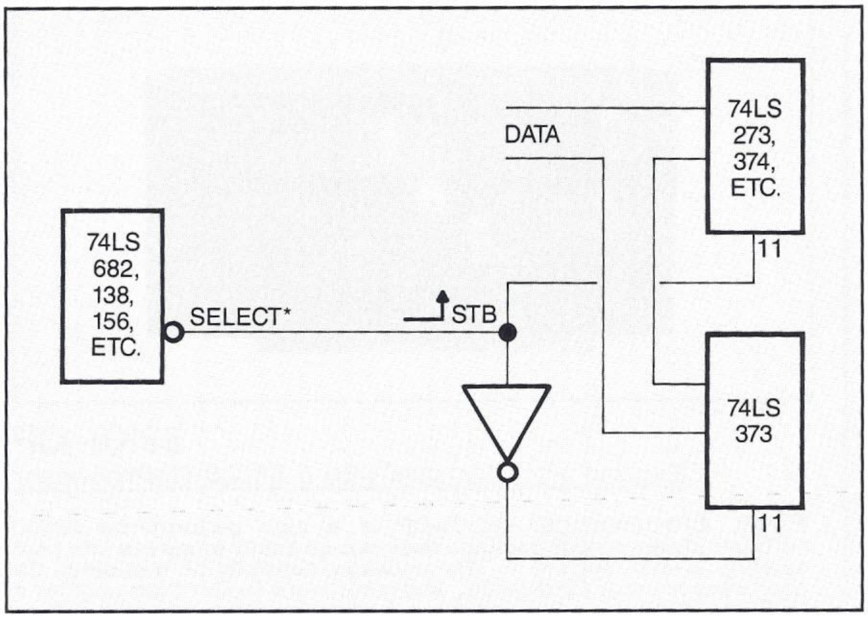

74LS
682,
138,
156,
ETC.
SELECT*
STB
DATA
74LS
273,
374,
ETC.
11
74LS
373
11

Figure 1. *Generating on-board strobes. This method stores the data present one setup time before STB (the end of the SELECT* pulse).*

flip-flops in series. They function when the SELECT* line goes true and are toggled by the system clock. This circuit has the advantage that we can set it up so that we know for sure that the data will still be valid when the strobe goes active. However, more components and board real estate are required. We should use this or some equivalent circuit when there is any doubt about the validity of the data at clock time.

CONCLUSION

It is certain that the S-100 bus will be upgraded to use the new 32-bit processors. Let us hope some standards are approved soon, so we do not run into the problems that we had in the early years of the S-100 bus. ■

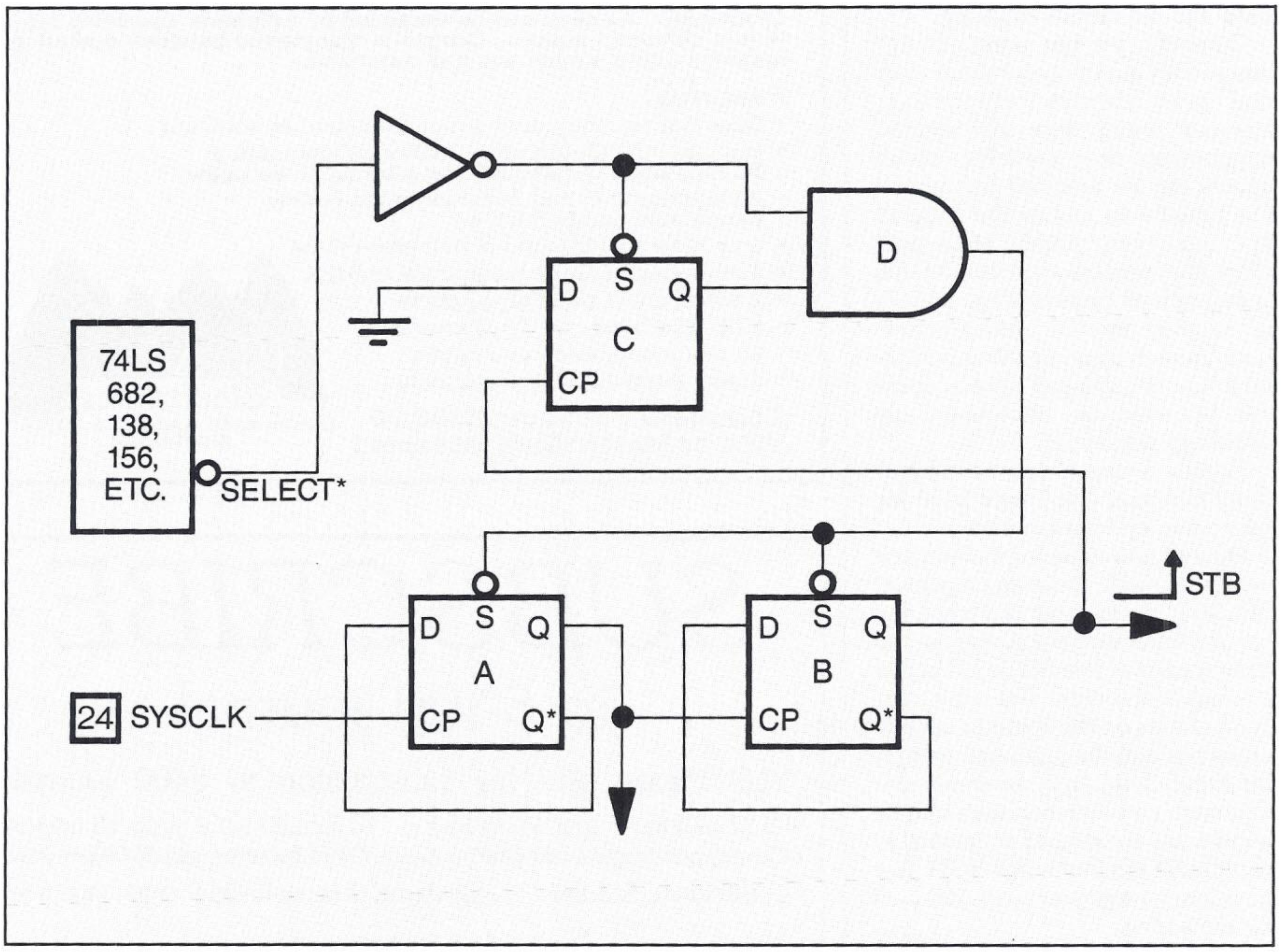

74LS
682,
138,
156,
ETC.
SELECT*
D
S
Q
C
CP
D
D
S
Q
A
CP
Q*
D
S
Q
B
CP
Q*
24 SYSCLK
STB

Figure 2. *Generating on-board strobes. Here, the Q output of flip-flop A will create a positive going edge with the second positive edge of Φ. The output at B will go high with the fourth clock pulse. C and D assure that only one positive edge is generated in any one bus cycle.*

WRITE FOR S-100 JOURNAL

You may have the information that many other S-100 Journal readers are looking for. Why not communicate it through an article in S-100 Journal?

We buy articles in a variety of subjects related to S-100 computers and operating systems running on S-100 computers. If you would like more information on writing for us, please request our Author's Guidelines. You may also wish to call our editor, Jay Vilhena, at 801-373-0696 to discuss any possible articles that you have in mind.

Among the many topics that we will consider for coming issues are LANs and other Multiuser Applications, System Integrations, Software Drivers for S-100 Boards, Specialized Applications, Zenith Z-100, and Upgrading Old Altairs and IMSAIs.

FOR SALE
USED S-100 EQUIPMENT

Ithaca and TEI buckets	$200 - $400
Z80 CPU boards	$25 - $75
64K RAM boards	$25 - $75
I/O boards	$25 - $75
19" rack mount cabinets	$100 - $300
Shugart disk drive (15 & 30 Mb)	$50
Hazeltine terminals	$50

— MANY OTHER ITEMS —

ALEXANDER TECHNICAL SERVICES
2252 Northwest Pkwy, Suite F
Marietta, GA 30067
404-980-1446

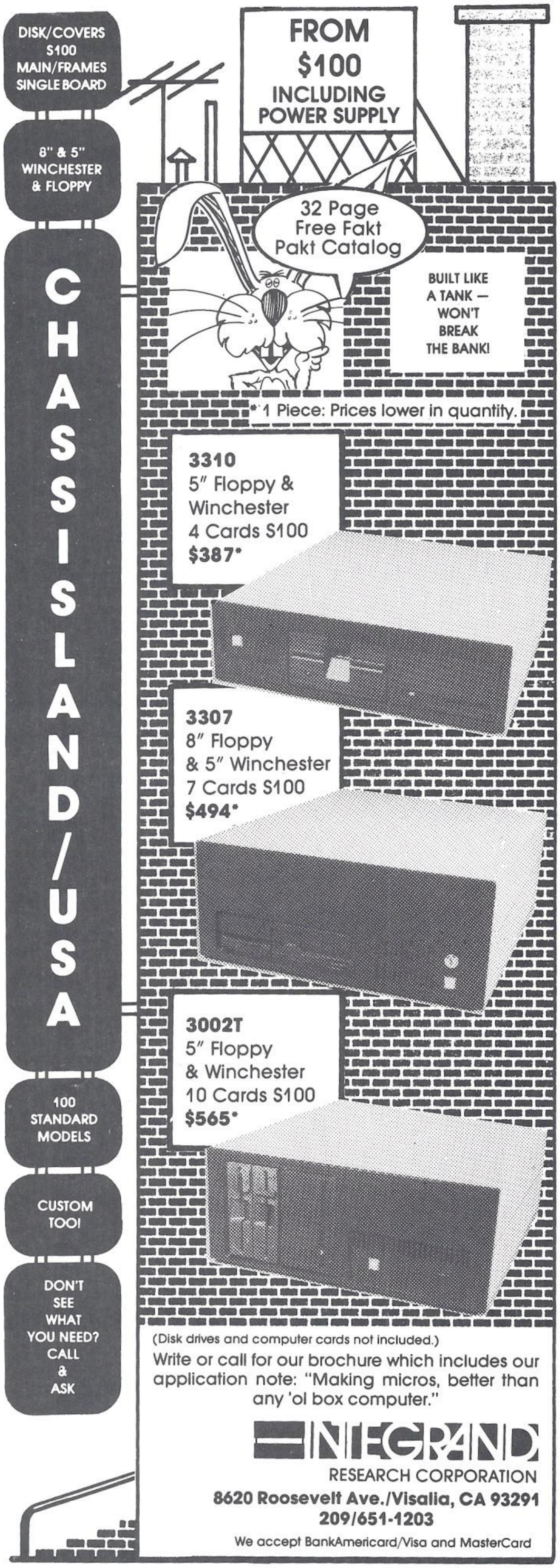
ALPHA MICRO USERS SOCIETY

You may know who we are, but do you know how we can help YOU???

AMUS has many services available specifically designed to help ALPHA MICRO USERS

Call or write for a free copy of our magazine and an outline of our services.

AMUS
735 Walnut Street
Boulder, Colorado 80302
(303) 449-6917



DISK/COVERS
S100
MAIN/FRAMES
SINGLE BOARD
8" & 5"
WINCHESTER
& FLOPPY
CHASSIS/SLAND/USA
100
STANDARD
MODELS
CUSTOM
TOO!
DON'T
SEE
WHAT
YOU NEED?
CALL
&
ASK
FROM
$100
INCLUDING
POWER SUPPLY
32 Page
Free Fakt
Pakt Catalog
BUILT LIKE
A TANK —
WON'T
BREAK
THE BANK!

*1 Piece: Prices lower in quantity.

3310
5″ Floppy &
Winchester
4 Cards S100
$387*

3307
8″ Floppy
& 5″ Winchester
7 Cards S100
$494*

3002T
5″ Floppy
& Winchester
10 Cards S100
$565*

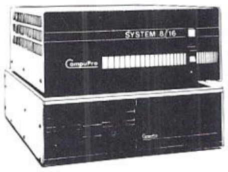
(Disk drives and computer cards not included.)

Write or call for our brochure which includes our application note: "Making micros, better than any 'ol box computer."

INTEGRAND
RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291
209/651-1203
We accept BankAmericard/Visa and MasterCard

CompuPro™

SYSTEM UPGRADES



- More Users
- New Operating System
- Larger, Faster Hard Disk
- PCs Networked to System
- Custom Database Software

Get it done right the first time

(415) 495-5999

COCHLIN COMPUTER SYSTEMS
330 TOWNSEND STREET #107 SAN FRANCISCO, CA 94107

BACK ISSUES OF S-100 JOURNAL

Back Issues are $4.90 each. We pay U.S. postage (for international orders, please add $2.00 per issue).

To get back issues, send this form or a note to:

S-100 Journal Back Issues
PO Box 1914, Orem, UT 84057

(Payment must be included with your order)

Please send S-100 Journal:

☐ NO. 1 ☐ NO. 2 ☐ NO. 3

NAME: ______________________________

____

ADDRESS: ______________________________

____

Enclosed is ____________ ($4.90 for each copy ordered. Plus $2.00 each for international orders.)

CROMEMCO'S XXU A 32-BIT 68020/68881 S-100 CPU BOARD

Ed Lupin
Cromemco

The new Cromemco XXU processor board provides a bold leap in microcomputer processing power by integrating a 16.7-MHz 68020 microprocessor, a 16.7-MHz 68881 math coprocessor, and a 16K associative cache on a single S-100 card. The XXU also contains a real-time, battery-backed-up clock/calendar, a 64K EPROM for system diagnostics and booting operations, and supports full 32-bit transfers on the S-100 bus.

HIGH-SPEED OPERATIONS

A key to the high performance of the XXU is the tight coupling of the 68020, 68881, and 16K bytes of high-speed cache memory on a single S-100 card. The cache memory on the XXU utilizes state-of-the-art CMOS RAM chips with a remarkable 25-nanosecond access time. Programs are executed out of cache memory with **zero wait states**. The 68020 is able to execute out of cache memory approximately 90% of the time. This unloads the S-100 bus and frees it for other operations (say DMA disk transfers) which can then occur in parallel with the CPU operation. When the cache is refilled from the bus, it is done with full 32-bit transfers from newer 32-bit cards (such as the Cromemco 2048KZ, a 2-megabyte memory card) or with 16-bit or 8-bit transfers from older cards.

The 68881 math coprocessor is able to execute either single-precision or double-precision floating point operations in parallel with the operation of the 68020. Not only can the 68881 do the basic arithmetic operations but a full set of transcendental and trigonometric functions as well. The 68881 is itself an 80-bit-wide processor, and it operates at a full 16.7 MHz clock rate (most 68881 coprocessors on the market today use the slower, 12.5-MHz part). If even higher speed versions of the 68881 are offered in the future, they too can be accommodated by the XXU. Provision is made for an optional crystal oscillator — just above the 68881 chip on the board — to provide a separate clock source for higher-speed devices.

For even further speed enhancement, the XXU implements a feature called *quick-write* to speed write operations to the S-100 bus. When the MC68020 performs a write cycle, the status, address, and data to be written to main memory are saved in registers external to the processor, and the MC68020 write cycle is terminated. The XXU state machine completes the write cycle to the S-100 bus, while the MC68020 continues execution from its internal cache or from the on-board external cache. By paralleling bus operations with cache, ten or more instructions can be executed in the time that would have been wasted had the MC68020 waited for the S-100 write cycle to be completed.

32 BITS ON THE S-100 BUS

To take complete advantage of the 68020 processor, the XXU supports full 32-bit transfers on the S-100 bus. The implementation was done in such a way that the XXU is completely compatible with existing byte-wide and word-wide slave devices meeting the IEEE-696 standard, and still it can take full advantage of newer 32-bit wide products. The protocol for ►

Ed Lupin is a Senior Computer Engineer with Cromemco and is the designer of the XXU card. He has been designing S-100 products for Cromemco for the past 7 years. Ed is a Higher Honors graduate of the University of Illinois.

Manufacturer	Model	Whetstones/sec.
CROMEMCO	CS-420	1,050,000
DEC	MICROVAX II	877,000
SUN	3/50	860,000
APOLLO	3000	780,000
DEC	VAX 11/780	476,000
IBM	PC/RT	200,000

Table 1. *Whetstone benchmark results. The Cromemco XXU 32-bit card was run on a System 420 with version 1.0 of Cromemco's 68020 Fortran compiler. Other data were obtained from Datamation and Unix World magazines.*

handling 32-bit transfers was modeled after the IEEE-696 convention for distinguishing 16-bit (word) transfers from 8-bit (byte) transfers. To achieve 16-bit transfers, the CPU makes a request on the sXTRQ* line of the bus. If the slave device (e.g., a memory board) responds with a SIXTN* signal on the bus, then a word-wide transfer is made. Otherwise two separate byte-wide transfers are made on two separate bus cycles.

In a similar way, the XXU can request a 32-bit-wide transfer on the sMURQ* line (for MUltiple ReQuest), defined as pin 69 of the S-100 bus. If the slave device is capable of 32-bit transfer, it responds with a MUAK* signal on pin 66 of the bus. The 32-bit transfer is then accomplished as two, rapid, successive 16-bit transfers in a **single** bus cycle. Using this technique, 32-bit data is transferred at an impressive rate of 8.33 megabytes per second on the S-100 bus. If the slave does not respond with MUAK*, then SIXTN* is checked and either 16-bit or 8-bit transfers are performed. This assures compatibility — although at lower transfer speeds — with IEEE-696 boards not capable of 32-bit transfers. *(Editor's note: Pins 66 and 69 are defined in the IEEE-696 standard as 'uNDEFined' and 'Reserved for Future Use' respectively. The signals sMURQ* and MUAK*, implemented by Cromemco, are not presently part of the standard).*

Manufacturer	Model	Dhrystones/sec.
CROMEMCO	CS-420	3546
CELERITY	C-1200	3468
PYRAMID	90x	3333
DEC	VAX 11/785	2136
GOULD	PN6005	1964
HP	HP 9000-500	1724
AT&T	3B20	1724
APOLLO	DN660	1666
DEC	VAX 11/780	1662
DEC	MICROVAX II	1612
IBM	PC/RT	1333
IBM	PC/AT	1315
AT&T	3B2	1315
PLEXUS	P/60	1163
DEC	VAX 11/750	1091
ALTOS	586	793
ONYX	C8002	511
IBM	PC/XT	427
IBM	PC	390

Table 2. *Dhrystone benchmark results. The Cromemco XXU 32-bit card was run on a System 420 with version 1.0 of Cromemco's 68020 C compiler. Other data were obtained from the UNIX USENET study.*

MANY FEATURES

The XXU also has many features designed specifically for operator convenience. A real-time clock/calendar is included on the XXU (using an Intersil ICM 7170 chip) so that the system always has access to time and date. This information is automatically obtained by the operating system; time and date need not be entered by the operator. The clock circuit is powered by a lithium-cell battery with a seven-year life, so accurate time and date information is maintained even when power is removed form the system.

The XXU has an on-board 64K EPROM containing a program we call XDOS. XDOS is used to boot the system and run a complete set of system diagnostics. By including the XDOS EPROM on the XXU card itself, the XXU is the only system card that needs to be functional in order to perform system diagnostics.

There is also a self-check capability, under the control of XDOS. On power-up, the XXU performs a complete self check and indicates any problem by means of a red, fault-detect LED on the XXU card.

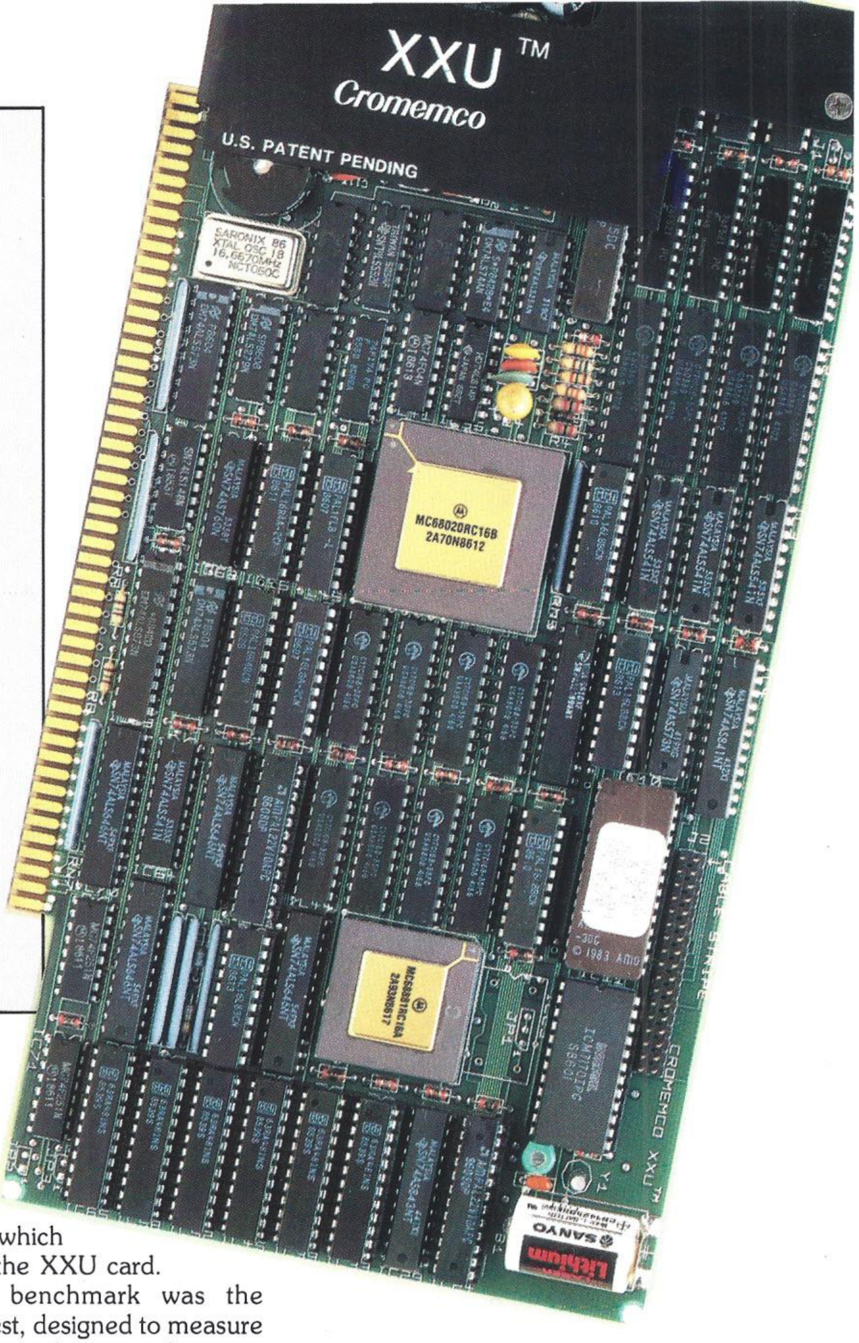
Memory management, required in UNIX systems, is also supported as an option by the XXU. The Cromemco memory management card, the XMU, couples to the XXU by means of a 34-conductor ribbon cable, interconnecting the top connector of the XXU with that of the XMU.

EXTENSIVE SOFTWARE

Two operating systems are supported on the XXU, Cromix Plus and UNIX System V. Both have been written

CROMEMCO XXU

USE:	Main Processor Board
MANUFACTURER:	Cromemco, Inc. PO Box 7400 Mountain View, CA 94039 (415)-964-7400
CPU:	Motorola 68020, 16.7 MHz
COPROCESSOR:	Motorola 68881, 16.7 MHz
CACHE MEMORY:	256-byte instruction cache (on chip). 16K 2-set associative instruction/data cache (on board).
FLOATING POINT:	IEEE Standard P754
DIAGNOSTICS:	64K XDOS EPROM Fault-detect LED
MEMORY ACCESS:	Longword (32 bits) Word (16 bits) Byte (8 bits)
CLOCK:	Real time (time and date) Battery backed (lithium)
POWER REQRMNTS:	+16 Volts at 1.8 Amps. On-board DC-to-DC supply produces 5V at 5 Amps.
LIST PRICE:	$4995



specifically for the 68020 processor, and optimized to take full advantage of the 68020. A complete set of programming languages is also available —including Basic, C, Pascal, and Fortran — which produce optimized 68020 code and take full advantage of the 68881 coprocessor for math operations.

Both the Informix and Unify data base packages are available from Cromemco as are the popular Ultracalc spreadsheet and the Quadratron office automation packages. In addition, the International Association of Cromemco Users (P.O. Box 17658, Irvine, CA 92713) has published a 200-page catalog of software available for Cromemco systems.

BENCHMARKS TELL THE STORY

To test the effectiveness of the XXU card, two industry-accepted benchmarks were run. The benchmarks were executed on a Cromemco System 420, Cromemco's newest super-micro which is based on the XXU card.

The first benchmark was the Whetstone test, designed to measure the computational power of a computer system. This benchmark was developed in England by H. J. Curnow and B. A. Wichmann and consists primarily of floating-point math and trigonometric functions. The results, shown in Table 1, indicate that the XXU-based system outperforms expensive minicomputers and offers five times the performance of even the most capable personal computer.

The second benchmark was the Dhrystone test (Table 2). This test, developed by R. P. Wicker, is designed to complement the Whetstone test by measuring the speed of data manipulation rather than numeric calculation. Again, the XXU-based System 420 is substantially faster than much more expensive minicomputer systems.

S-100 BUS: A NEW ERA

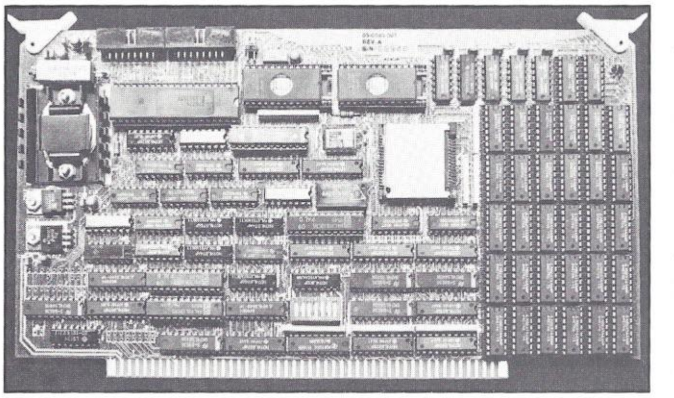
It was just 10 years ago that Harry Garland and Roger Melen, founders of Cromemco, coined the term *S-100 bus* and introduced the first S-100 Z-80 processor card (the ZPU). Measured by Whetstone performance, the ZPU clocked in at 7000 whetstones per second. Based on this measure, the XXU has **150 times the performance** of the ZPU! With the XXU card, S-100 systems can clearly outperform even some of the most capable minicomputers, and do so at a price/performance ratio unequalled by any other technology in the industry. ■

news and new products

IBS Introduces 10-MHz 80186 Slave Board

Independent Business Systems has announced the Slavenet 186/512. This 16-bit S-100 slave board features an 80186 CPU running at 8 or 10 MHz, 2 serial ports, and up to 1 megabyte of dynamic RAM. The board is capable of true 16-bit data transfers at 2 megabytes per second. The price ranges from $995 for a 8-MHz 256K board to $1512 for a 10-MHz 1-Meg board.

IBS has also introduced two new S-100 motherboards available in either 12- or 20-slot configurations. Both motherboards include a battery-backed clock/calendar, are actively terminated, and are designed to reduce RF interference on the bus. List price is $250 and $300 respectively.



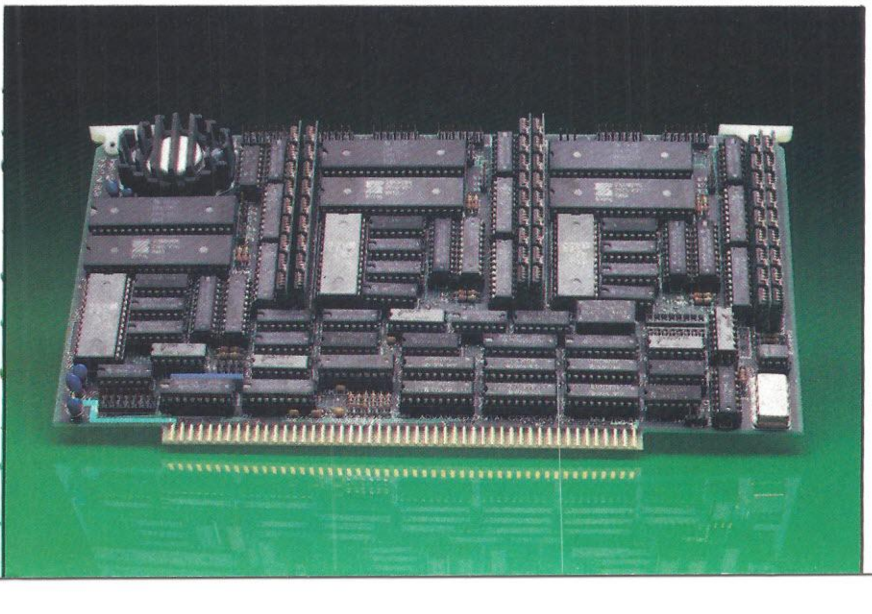
IBS products carry a 3-year warranty. For more information contact or send an Editorial Feature Reply Card to Independent Business Systems, Inc., 5915 Graham Ct., Livermore, CA 94550. ★

Triple Z80H Slave Board From Advanced Digital

Advanced Digital Corporation has introduced the Multi-Slave, a 3-user, 3-processor board compatible with TurboDOS. The Multi-Slave features 128K of RAM and 2 serial ports for each of the three users. The Z80H processors run at 8 MHz. The retail price for this IEEE-696 card is $695.

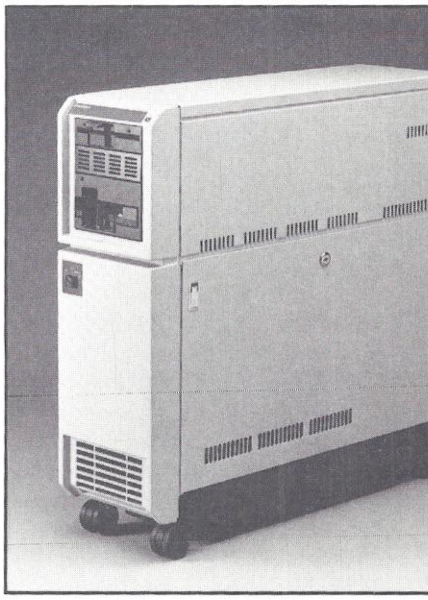
Also from Advanced Digital comes the Super 16, a 16-bit slave board featuring an 80186 CPU running at 8 MHz. The Super 16 contains 256K of RAM, 4 serial ports, and runs under TurboDOS.

For more information on these products, contact or send an Editorial Feature Reply Card to Advanced Digital Corporation, 5432 Production Drive, Huntington Beach, CA 92649. ★



Cromemco Produces 32-Bit 68020 S-100 Supermicro

Cromemco, Inc. has packaged its new 32-bit XXU card into a supersystem called the CS-420. Running at 16.7 MHz, the 68020/68881 XXU card confers the CS-420 a level of performance that, the company claims, surpasses that of many popular minicomputers. The CS-420 includes 32-bit memory boards and can accomodate up to 16 megabytes

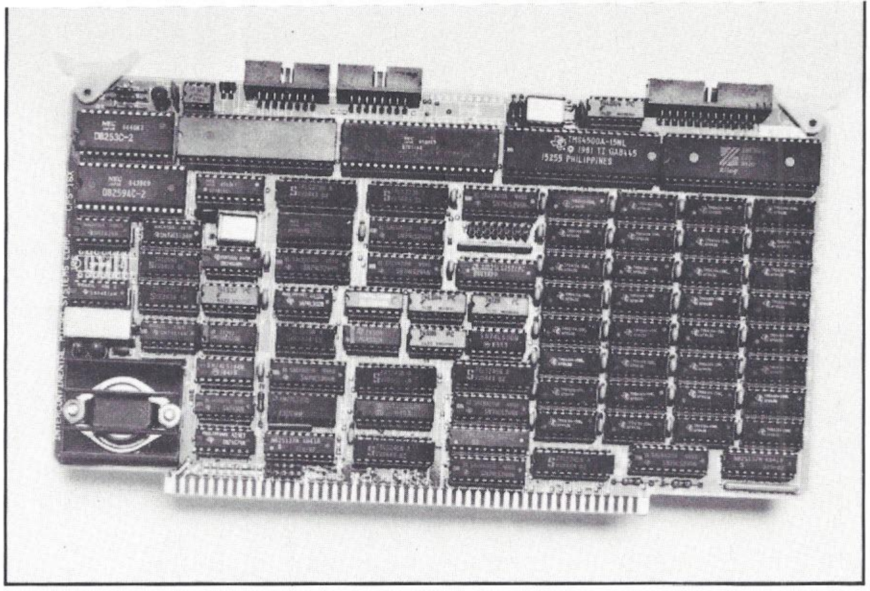


of RAM. It can serve up to 64 users under the UNIX V.2 operating system.

A unique option for the CS420 is the Cromemco S-series line of graphics products. With this option, the CS-420 can capture, store, create, and display high-resolution television images.

The CS-420 includes 5¼ floppy drive, 32-Megabyte cartridge tape drive, and up to 280 megabytes of hard disk storage. Prices start at under $28,000. For additional information contact or send an EF Reply Card to Cromemco, Inc., PO Box 7400, Mountain View, CA 94039. ★

S-100/TurboDOS Package From ICM Emulates IBM-PC



From InterContinental Microsystems is available the MS-1000, a special package for the S-100 bus that will allow running PC-DOS and MS-DOS monochrome software under TurboDOS. The package consists of the MS-1000 software, an S-100 board, an IBM-compatible terminal, PC-DOS 3.1, and TurboDOS/PC software.

The S-100 board included is the new CPS-16F, a network processor board featuring a 10-MHz NEC V30 microprocessor. The V30 CPU runs both 8086 and 8080 programs for 8-bit or 16-bit applications. This board also features one megabyte of RAM, two serial ports, two parallel ports, and a real-time clock.

The full package lists for under $2000. To obtain complete information write or send an Editorial Feature Reply Card to InterContinental Micro, 4015 Leaverton Court, Anaheim, CA 92807. ★

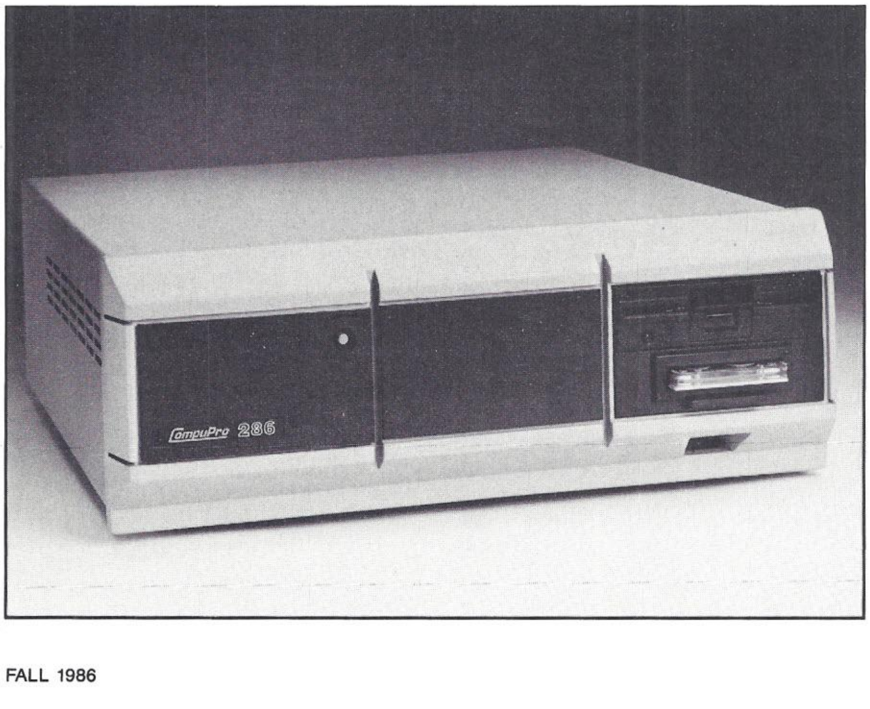
80286-Based Multiuser CompuPro System Available

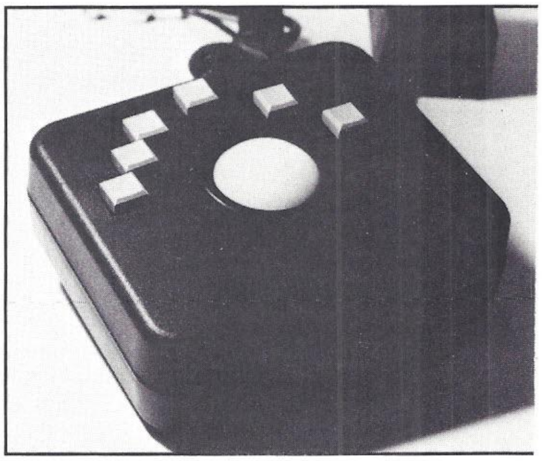
Viasyn Corporation has begun volume shipments of the company's new top-of-the-line system, the CompuPro 286/80. The system is built around the Intel 80286 CPU and runs under CompuPro's Concurrent DOS 816. The 286/80 includes an 80-Megabyte hard disk with a dedicated MDrive/H RAMdisk board functioning as a cache buffer.

The CompuPro 286/80 comes with a 16-slot motherboard and has a rear panel with room for up to 18 serial connectors.

Other features included are 9 serial ports, built-in tape backup unit, 5¼ floppy drive, and 768K of system memory. Options include slave processors, LAN interface boards, graphics, and additional mass storage. The 286/80 can also be used as a file server to other computers.

For more information, contact or send an EF Reply Card to CompuPro, 26538 Danti Court, Hayward, CA 94545-3999. ★





Unique Cursor Pointing Device From Fulcrum

Fulcrum Computer Products has released a trackball pointing device, the Trackball Plus. This hand-driven accessory emulates most popular mice and digitizer formats, such as Summagraphics Bit Pad One, Houston Intruments Hipad, Tektronix Plot-10, Mouse Systems, and others. But, according to Fulcrum, the Trackball Plus offers many advantages over mice; for example, it only requires 4.62″ by 5.37″ of desk ►

space and can be moved simply by fingertip motion.

Six user switches allow easy selection of screen menu options and cursor positioning, including alternate cursor. Trackball Plus interfaces to your S-100 computer through a regular RS232 serial interface. Retail price is $95. Contact or send EF Reply Card to Fulcrum Computer Products, 451 Allan Ct., Healdsburg, CA 95448.

MULTIPLE BAUD-RATE MODEM AVAILABLE FROM FASTCOMM

Fastcomm Data Corp. introduced a modem capable of transmitting at 300, 1200, 2400, and 9600 baud. It is available in several designs.

The FASTCOMM modem features auto-dial and auto-answer, supports a superset of Hayes-compatible commands, and allows monitoring of transmissions through external LEDs and a speaker. Price is around $1000.

For more information contact or send EF Reply Card to Fastcomm Data Corp., 12347-E Sunrise Valley Drive, Reston, VA 22091. ★

SOFTWARE

From MultiComputer Technology Corporation is now available version 1.42 of The Menu System. This software encases all of TurboDOS commands in screen-oriented menus. A hierarchy of menus can be user-defined. Cost is $59.95. MultiComputer Technology Corporation, 126 Northpoint, Suite 152, Houston, TX 77060. ★

S-100 Microsystems has developed a set of programs, the CTU package, that allow using the Alloy cartridge tape subsystem on CompuPro 68K systems running UniSoft's UniPlus+ version of Unix system V. The software offers tape positioning and read/write functions. No reconfiguration of the operating system is required. Price is $395. Source code is also available for $995. From S-100 Microsystems, 1337 Heidi Drive, Plano, TX 75023. ★

For Alph Micro users, the Simplifile data base system is now available from Clark Associates, Inc. Simplifile allows data fields to be added, deleted, or expanded and file creation and modification. It is disk-based to save memory and features user-definable report generator and password security. The package lists for $495. Clark Associates, Inc., 1610 E. Algonquin Road, Schaumburg, IL 60195. ★

Cleydale Engineering offers a set of utilities for the Microsoft FORTRAN-80 compiler running under CP/M-80. The package includes an optimized scientific subroutine library. FORLIB.REL math additions, peripheral device drivers, and FORTRAN programming tools. Functions available are regression analysis, matrix operations, equations, fast Fourier transforms, integration, graphics, celestial mechanics, program source line renumbering, file scrolling, and many others. A total of 60 files. Cost is $49.95. From Cleydale Engineering, Rt. 1 Box 217-B, Blacksburg, VA 24060. ★

BOOKS AND CATALOGS

A catalog of software for North Star and 8-inch CP/M computers is available free upon request from Dynacomp, Inc., 1064 Gravel Road, Webster, NY 14580. This catalog lists hundreds of software packages, most reasonably priced, and even includes lots of public domain titles. ★

For those planning to interface their S-100's to the IEEE-488 bus, a free catalog of interface boards and accessories is available from National Instruments, 12109 Technology Boulevard, Austin, TX 78727-6204. ★

Users and future users of the Z Operating System can now obtain the Z-System User's Guide, by R. Jacobson and B. Morgen. The book includes tutorials and examples on using Menus, Pathes, Named Directories, Aliases, Shells, etc. 90-pages, $14.95 plus shipping. From Echelon, Inc., 885 N. San Antonio Road, Los Altos, CA 94022. ★

NEWS

MACROTECH International has acquired all rights and all the S-100 inventory of the defunct Octagon and S/D Systems companies (later called Syntech Data Systems). Now all owners and users of the Versafloppys and Expandorams are no longer left in the cold since MACROTECH is offering tech support on the products. For support or information on this line of products, contact MACROTECH International Corp., 21018 Osborne Street, Unit #5, Canoga Park, CA 91304. (818) 700-1501. ★

Looks like Alpha Microsystems will now become part of Televideo. Alpha has been losing money lately and apparently is no longer capable of operating on its own. Notice that Alpha Microsystems, a company that grew by selling S-100 systems, has been slowly moving away from the S-100 line and never really adhered completely to the IEEE-696 standard. Fair warning! ★

If your desire is to know all about Optical Disks and CD-ROMs, you may want to attend the Optical Information Systems '86. This conference takes place December 9-11 in Arlington, Virginia. Prices vary from about $100 to about $500 depending on how many days and sessions you attend. The conference features 13 sessions, special workshops, and over 60 exhibitors of optical media products. Contact Conference Management Corporation, 17 Washington Street, PO Box 4990, Norwalk, CT 06856. (203) 852-0500. ■

The News and New Products *section is compiled from press releases and other information supplied mostly by product manufacturers. Information about S-100 products has first-priority when considered for publication. High-quality photographs are welcome. Press releases should be dated and should include prices. Send to New Products, S-100 Journal, 1275 N. University Ave., Unit 7, Provo, UT 84604*

s-100 software

$20 PER ISSUE BUYS THIS SPACE

List your software product here in the S-100 Software directory. All listings must be of software that runs on an S-100 system. Ads of software that runs only under MS/PC-DOS will not be accepted. Messages must have approx. 30 words or less. We do all typesetting free.

DataCOM Super Systems™

(813) 791-1454 modem 300/1200/2400 bps

Remote multi-user 140mb TurboDOS system with files for TurboDOS, CP/M, MP/M, and PC/MS-DOS users. Courier 2400 modems and 2 phone lines. $35.00 annual fee. Call & download application today.

CO-SY, COMPUTER UND SYSTEME

Krischerstr. 70, d-4019 Monheim, West-Germany
(2173) 52071 or 72

We are looking for cooperation with Software manufacturers-authors to adapt and translate Software into German. We specialize in UNIX and C language, MS-DOS and C, and C-Basic.

NIRVONICS, INC.

PO Box 5062, Plainfield, NJ 07061
(201) 561-2155

Software for CP/M and MS-DOS. Products include: UNIX-style utilities, database program, form completion tool and an arcade-style action game requiring no graphics hardware.

Orders for custom software taken.

SCS

166 Wright Street, Staten Island, NY 10304
(718) 442-0665

DISMAL: table-driven disassembler you can configure for almost any processor and mnemonics. Preconfigured for Z80, all documentation and source included for total user customization. Specify disc format. $50 prepaid.

MultiComputer Technology Corporation

The Multiuser Software Company

126 Northpoint, Suite 152, Houston, TX 77060
(713) 847-0333

Software for TurboDOS systems. Business Financial Management. Dental Office Management. The Menu System.

See our ads in pages 15 and 22.

Affordable Engineering Software

CALL OR WRITE FOR FREE CATALOG

Circuit Analysis • Root Locus • Thermal Analysis • Plotter Drivers • Graphics • Signal Processing • Filter Design • Report Proofreader • Transfer Function Analysis.

BV Engineering

2200 Business Way, Suite 207, Riverside, CA 92501
(714) 781-0252 VISA/MC

$140 BUYS THIS SPACE FOR TWO YEARS

That's only $17.50 per issue. All listings must be of software running on S-100 systems and must **not** run only under MS/PC-DOS. Messages must be 30 words or less. We do all typesetting free. To qualify for 2-year rate, please send $140 with the text for your first listing.

HOWARD PETERS

Box 2127, Rockford, IL 61131

JOB — the best CP/M tasking utility. Replaces old submit/xsub. **JOB** runs virtually ALL programs — xsub won't! Many other capabilities — keyboard entry, nesting, wildcard, list driven, IF, GOTO, more. Customize it — Turbo/ASM SOURCE CODE and utilities included! Information SASE. Program $15.00, 8 inch SSSD. Inquire about 5 inch formats.

ALL MULTIPLE-RUN ADS APPEARING IN THE S-100 SOFTWARE OR S-100 DIRECTORY CAN BE CHANGED FROM ISSUE TO ISSUE. Simply make certain that we have your text for the next issue no later than 2 months after the last one is out.

Due to the very low price of these ads, payment must be made in advance and no free sample issue will be sent. Send your listings to S-100 Journal, S-100 Software, PO Box 1914, Orem, UT 84057.

s-100 directory

$20 PER ISSUE BUYS THIS SPACE

List your business or product in the S-100 Directory. All listings must be S-100 related. Messages must have 30 words or less. We do all typesetting free. Send listing w/ $20 to S-100 Journal, S-100 Directory, Box 1914, Orem, UT 84057.

COCHLIN COMPUTER SYSTEMS INC.

330 Townsend St. #107, San Francisco, CA 94107
(415) 495-5999

Authorized CompuPro System Center. On-site service and maintenance, custom software design, multiuser system integration, local area networking, user training and more.

MENTZER COMPUTER SYSTEMS

1441 Rollins Rd., Burlingame, CA 94010
(415) 340-9363

CompuPro Full Service System Center. Custom S-100 systems for Accounting, Medical, and Legal applications. Repair and maintenance of all brands of S-100 systems.

MICRO POWER UNLIMITED

1716 Erring Rd. #102, Simi Valley, CA 93065
(805) 584-6789

Authorized CompuPro System Center. We provide on-site service and maintenance, custom software design, multiuser system integration, user training and more.

JURIS SCRIVENER, INC.

2019 B Hollywood Blvd., Hollywood, FL 33020
(305) 920-7772

System Integrator and Vendor. S-100 Granturbo Computers running TurboDOS 8 and 16 bit, will also run CP/M and MS-DOS. Systems for 4 users and up. FMS 80 Data Base Management Software for S-100 computers. Primage printers.

SMALL BUSINESS SYSTEMS, INC.

PO Box 635, 9537 Jackson Ave
Brookfield, IL 60513
(312) 485-8338

CompuPro multi-user systems, networking, peripherals, software. Fast, reliable, and versatile systems for any business computer needs.

RENARD SYSTEMS INC.

4248 Ridge Lea Road, Amherst, NY 14226
(716) 833-4758

Service offices in NYC and Rochester. Full Service CompuPro System Center specializing in word processing automation, AutoCad on S-100, laser printers, very large hard disk drives, and tape backup. Service contracts to entire North East (out of NYC). National distributors for a variety of system enhancing software products, including PrintMan, Phantom Printer, Smart-Op and Speed-Op by Data Base Administrators.

ELECTRONIC SYSTEMS, INC.

7645 Production Drive
Cincinnati, OH 45237
(513) 761-5610

Your Midwest Source For CompuPro Multi-User Computers! On-Site Maintenance, Custom Software Systems, Peripherals. In Business Since 1958!

LPI - Micro Drive Service

10020 Fontana, Overland Park, KS 66207
(913) 648-1373

Unique service for the unique Micropolis disk drives found in Vector Graphic System B, Exidy Sorcerer, and other S-100 systems. Factory qualified, fast turnaround, reasonable prices.

MEGATECH

PO Box 4213, Yalesville Station, CT 06492
(203) 634-4593

We buy, sell, and trade S-100 boards. An enormous selection of backplanes, S-100 cards, and whole systems is available. Call or write for free catalog. Complete repairs also available.

PMMI S-100 MODEMS Professionally Serviced

Guaranteed 90 days private use. 300/600 Baud. Paid Orders from stock. No CC. Order PMMI/103MS, Send $54.45 incl P&H to

Sparks Micro, Box 880, Sparks, MD 21152

S-100 Specialists Since 1978.

$140 BUYS THIS SPACE FOR TWO YEARS

That's only $17.50 per issue. All listings must be S-100 related. Messages must be 30 words or less. Message can be changed for each new issue (notify us no later than 2 months after previous issue). We do all typesetting free. Send first listing with $140 to S-100 Journal, S-100 Directory, PO Box 1914, Orem, UT 84057.

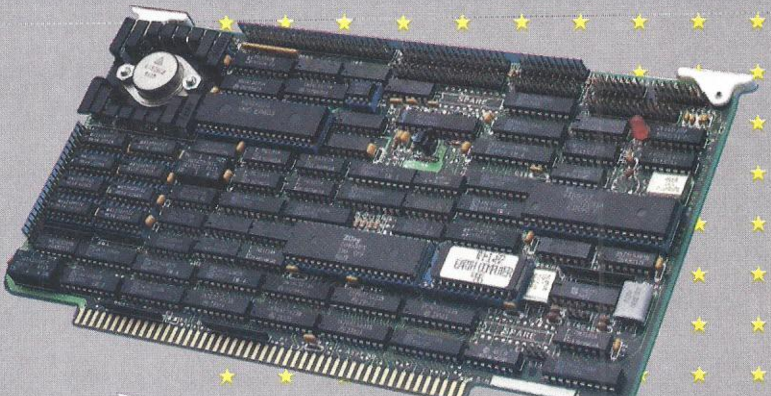
EARTH LAUNCHES NEW STARS

EARTH COMPUTERS launches some of the 'Hottest' new stars in the S-100 Universe. EARTH's new stars are fully compatible with the Multiuser TurboDOS operating system, and will operate in most S-100 systems, including pre-IEEE 696 systems.

TURBOMASTER 8

This outstanding new 8-Bit Single Board Computer offers features that are out of this world:

- On-board ST-506 Winchester Controller
- TurboDOS, CP/M, MP/M compatible
- 5¼″ and 8″ Floppy Controller
- Up to 256K of memory
- 8-MHz, Z-80H CPU
- 2 serial ports, 1 Parallel port



TURBOSLAVE I

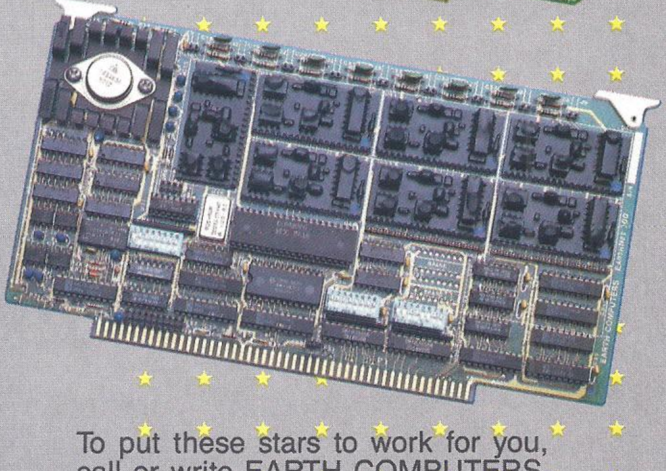
The perfect companion to the TURBOMASTER 8 or other 8/16-Bit Master processor. This high-speed slave utilizes an 8-MHz Z-80H CPU and offers extensive on-board diagnostics... an industry exclusive.

- No paddle boards
- 128K of RAM
- 2 RS-232 ports, 50-38.4 KBaud
- FIFO communications
- Data transfers to 1 MByte

EARTHNET S-100 and EARTHNET PC

The low cost, ARCNET-compatible way to tie S-100 systems and PCs.

- Token-passing architecture
- 2.5 Megabit data rate
- 255 nodes per network
- 2000 feet between nodes
- Simple coax wiring
- Optional on-board active hub

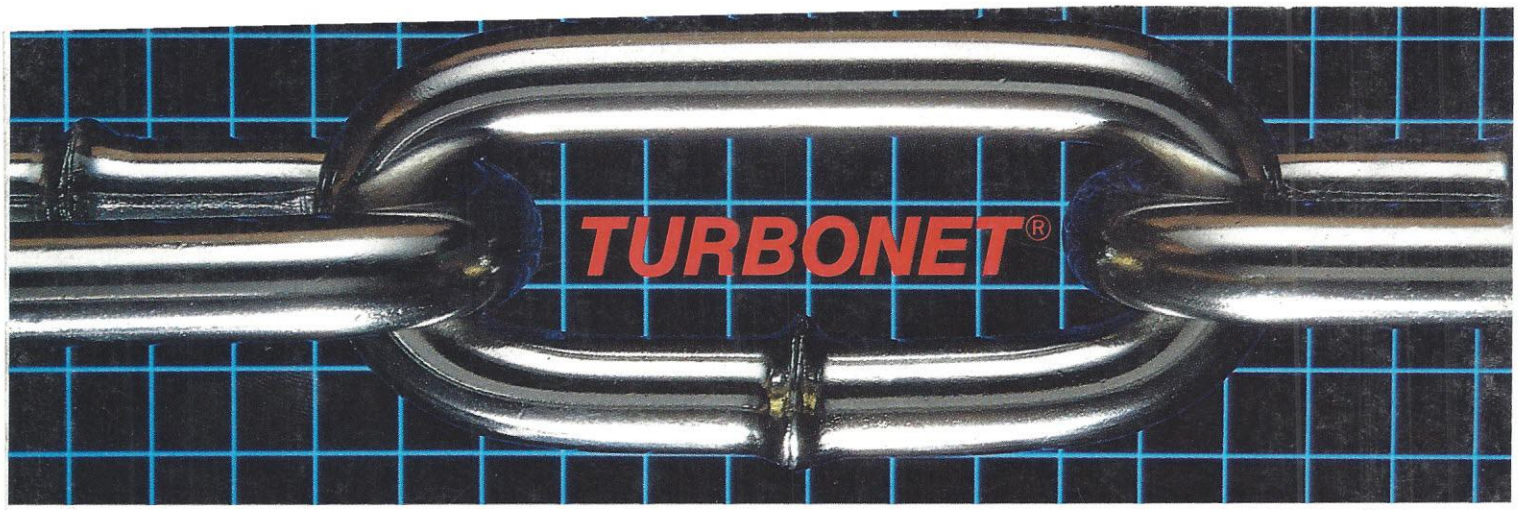


To put these stars to work for you, call or write EARTH COMPUTERS.
BE SURE TO ASK HOW YOU CAN WIN A FREE Z-80 COPROCESSOR BOARD.

EARTH COMPUTERS

"Building Blocks For The Super Micro"

P.O. Box 8067, Fountain Valley, CA 92728 • TELEX: 910 997 6120 EARTH FV • PHONE: (714)964-5784



One Strong Link Can Strengthen Your Whole System

TurboNET® is Teletek's new local area network that permits reliable high speed data transfers among computers of differing bus architectures. TurboNET is designed to be used in networks consisting of multiple S-100 based and PC based systems. It will allow up to 4000 users, including up to 255 IBM PCs or compatibles, to share a single network and all attached peripherals. The network can be organized in any number of different ways mixing Teletek's 8 and 16-bit multiuser systems and PCs in any combination.

Teletek's Networking Family consists of:

TurboNET PC:
IBM-PC Network Interface Board

Teletek's TurboNET PC board offers IBM-PC Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

TurboNET S-100:
S-100 Network Interface Board

Teletek's TurboNET S-100 board offers IEEE 696 Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

The benefits are obvious: The cost savings of shared peripherals, almost unlimited system expansion capability, and the use of existing PC workstations with the ability to run the myriad of application software written for MS-DOS and PC-DOS. This coupled with Teletek's 8 and 16-bit multiuser systems running application software written for CP/M and MP/M allows the system the ability to access almost any software library.

For more information on Teletek's TurboNET S-100 and TurboNET PC boards or on any of our full line of S-100 products, please call our Sales Department at 916-920-4600.

TELETEK
TELETEK
TELETEK

TELETEK

4600 Pell Drive, Sacramento, CA 95838
(916) 920-4600 Telex #4991834 Answer back — Teletek

© 1986 Teletek